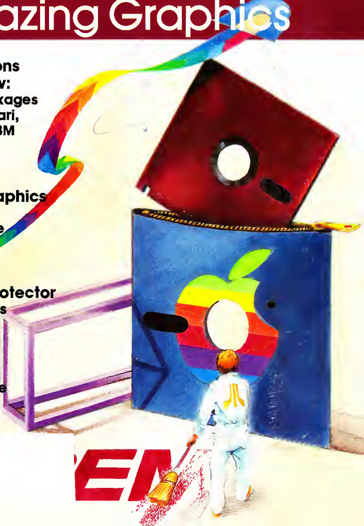
COMPUTE!

$2.95
November
1985
Issue 66
Vol. 7, No. 11
$3.75 Canada
02193
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

Amiga's Amazing Graphics



Telecommunications Software Overview:
72 Inexpensive Packages For Commodore, Atari, Apple, Macintosh, IBM

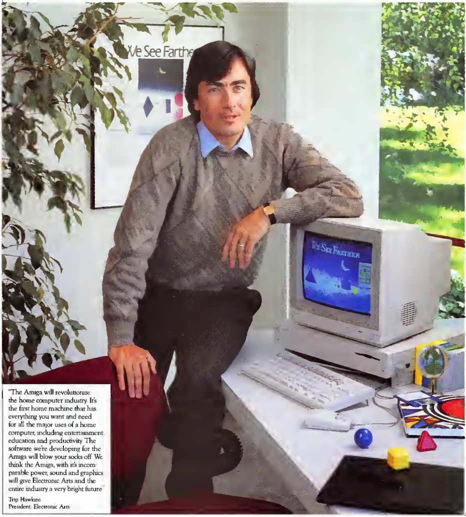
Commodore 64 3-D Animated Graphics
Simple Commands To Create And Move Your Own Shapes

Apple Program Protector
Shield Your Programs From Prying Eyes

Atari REMover
Automatically Delete REMs From BASIC

A message from a leading software publisher.

WHY ELECTRONIC ARTS



"The Amiga will revolutionize the home computer industry. It's the first home machine that has everything you want and need for all the major uses of a home computer, including entertainment, education and productivity. The software we're developing for the Amiga will blow your socks off. We think the Amiga, with it's incomparable power, sound and graphics will give Electronic Arts and the entire industry a very bright future."

Trip Hawkins
President, Electronic Arts

IS COMMITTED TO THE AMIGA.

In our first two years, Electronic Arts has emerged as a leader of the home software business. We have won the most product quality awards—over 60. We have placed the most *Billboard* Top 20 titles—12. We have also been consistently profitable in an industry beset by losses and disappointments.

Why, then, is Electronic Arts banking its hard won gains on an unproven new computer like the Amiga?

The Vision of Electronic Arts.

We believe that one day soon the home computer will be as important as radio, stereo and television are today.

These electronic marvels are significant because they bring faraway places and experiences right into your home. Today, from your living room you can watch a championship basketball game, see Christopher Columbus sail to the New World, or watch a futuristic spaceship battle.

The computer promises to let you do much more. Because it is interactive you get to participate. For example, you can play in that basketball game instead of just watching. You can actually be Christopher Columbus and feel firsthand what he felt when he sighted the New World. And you can step inside the cockpit of your own spaceship.

But so far, the computer's promise has been hard to see. Software has been severely limited by the abstract, blocky shapes and rinky-dink sound reproduction of most home computers. Only a handful of pioneers have been able to appreciate the possibilities. But then, popular opinion once held that television was only useful for civil defense communications.

A Promise of Artistry.

The Amiga is advancing our medium on all fronts. For the first time, a personal computer is providing the visual and aural quality our sophisticated eyes and ears demand. Compared to the Amiga, using some other home computers is like watching black and white television with the sound turned off.

The first Amiga software products from Electronic Arts are near completion. We suspect you'll be hearing a lot about them. Some of them are games like you've never seen before, that get more out of a computer than other games ever have. Others are harder to categorize, and we like that.

For the first time, software developers have the tools they need to fulfill the promise of home computing.

Two years ago, we said, "We See Farther." Now Farther is here.

ELECTRONIC ARTS™

Dr J and Larry Bird Go One-On-One

The number one software sports game of all time. Shoot as accurately as Larry Bird, slam dunk like the Doctor, while you're cheered on by the victory chants of the Boston Garden crowd.

Skyfox™

Get in the spaceship and fly. Out your window or on your radar screen you have but split-seconds to appreciate the fierce beauty of enemy jets and tanks.

Seven Cities of Gold™

Be Christopher Columbus and discover the New World. Learn history and geography, or generate your own random new worlds to explore.

Archon

A new kind of computerized board game, like chess with wizards and dragons for pieces. But when one lands on another they have to fight a white-knuckled action battle.

Deluxe Video Construction Set™

Be your own video director for business presentations or just for fun. Set up special effects, animated computer graphics, sound effects and titles—even record them to videotape for use with a VCR.

Arcticfox™

You command the advanced and deadly tank of the future—the Arcticfox. A first person tank combat game with all the stunning graphics and sound of the best 3-D simulations.

Return to Atlantis™

Play Indiana Cousteau, oceanic hero, in this three dimensional simulation under the seven seas.

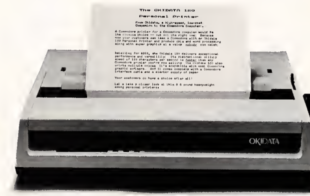
Marble Madness™

For the first time, the home version of a coin-op arcade game is just as good as the original. Same graphics. Same sound. And you can play it in your bathrobe.

For details about availability, see your Amiga software dealer or call us at (415) 572 ARTS. For a product catalog send $.50 and a stamped, self addressed envelope to: Electronic Arts, Amiga Catalog Offer, 2755 Campus Drive, San Mateo, CA 94403. Amiga is a trademark of Commodore Business Machines. Skyfox, Seven Cities of Gold, Deluxe Video Construction Set, Arcticfox, Return to Atlantis and Electronic Arts are trademarks of Electronic Arts. Marble Madness is a trademark of Atari Games, Inc.

See Dealer Listing on Page 88.

Left Brain.



Rational. Functional. Precise.
Introducing the OKIDATA 120, the logical printer for your Commodore® computer.

Get results fast. With a utility mode that zips through letters and reports at twice the speed of any Commodore printer.

Switch to the enhanced mode. And print your most important ideas with typewriter clarity. Or illustrate your rationale with the 120's bit image graphics for high resolution charts, graphs and drawings.

Stay on target. With a self-inking "Clean Hands" ribbon cartridge. And Okidata's famous full year warranty on parts, labor and printhead.

The OKIDATA 120. At $269*, it's the only Commodore-compatible printer that makes sense.

For more information, call 1-800-OKIDATA (in New Jersey 609-235-2600). Mt. Laurel, NJ 08054.

OKIDATA®
an OKI AMERICA company
We put business on paper.

Commodore is a registered trademark of Commodore Business Machines, Inc.
OKIDATA is a registered trademark of OKI AMERICA, INC.
*Manufacturer's suggested retail price.

Get the OKIDATA 120
at these fine stores:
Advantage Computer
Accessories (Canada)
Childworld/Children's Palace
Consumers Distributing
Crazy Eddie
David Weis
Electronics Boutique/
Games & Gadgets
Federated
Fred Meyer
Lionel/Kiddie City/
Lionel Playworld
Montgomery Ward
(at participating stores)
S. E. Nichols
Service Merchandise
Toys 'R Us

Right Brain.

Effervescent. Colorful. Outrageous.

Meet the OKIMATE 10, the $208* color printer that takes your Atari® or Commodore® computer over the rainbow!

Dazzle 'em. With brilliant printing in over 36 eye-tickling colors. Reds, greens, golds and blues that breathe life into everything: from charts and graphs to original drawings and overhead transparencies.

And when you're forced to work in black and white, the OKIMATE 10 prints crisp, clean reports and papers—at 240 words per minute. You can even add spice with wide, bold and fine print.

Everything you need for color printing comes with the OKIMATE 10 and its Plug 'n Print package. Including a data cable, Learn to Print and Color Screen Print software diskettes, a color ribbon cartridge, a black ribbon cartridge and paper.

So c'mon, print on the wild side. With the OKIMATE 10 Personal Color Printer from Okidata.

For more information, call 1-800-OKIDATA (in New Jersey 609-235-2600). Mt. Laurel, NJ 08054.

OKIDATA®
an OKI AMERICA company

We put business on paper.

*Manufacturer's suggested retail price Atari is a registered trademark of Atari, Inc
Commodore is a registered trademark of Commodore Business Machines, Inc.
OKIDATA is a registered trademark of OKI AMERICA, INC
OKIMATE and Plug 'n Print are trademarks of OKI AMERICA, INC
To run Plug 'n Print software, the Commodore 64, 128 and PLUS 4 require disk drive.
Atari requires disk drive and a 48K memory

**Buy Now!
$15 Manufacturer's rebate on OKIMATE 10.**
Offer good from October 1, 1985 through January 31, 1986. See the following participating stores for details.

Advantage Computer Accessories (Canada)
Arthur's Jewelers
Best Catalog Showrooms
Brendle's
Caldor
Childworld/Children's Palace
Consumers Distributing
Crazy Eddie
David Weis
Dolgins Catalog Showrooms
Electronic Boutique/ Games & Gadgets
Ellman's
Evans
Federated
Fred Meyer
Fred P. Gattas
G.C. Murphy/Murphy Mart
G.I. Joe
Great Western Catalog Showrooms
J. Triesman
Jafco Catalog Showrooms
LaBelles Catalog Showrooms
Lionel/Kiddie City/ Lionel Playworld
McDade
Meijers (Michigan only)
Montgomery Ward (at participating stores)
Prange
Save-Rite
Sears, Roebuck & Co. (at participating stores)
Service Merchandise
Stereo Village
Stokes
Toys 'R Us
Videoland
Witmark
Wizards Electronics
Zayre

COMPUTE!

NOVEMBER 1985
VOLUME 7
NUMBER 11
ISSUE 66

FEATURES

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 139 before typing in programs.

AP Apple **Mac** Macintosh **AT** Atari **V** VIC 20 **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16 **128** Commodore 128, **P** PET/CBM **TI** Texas Instruments **PC** IBM PC **PCjr** IBM PCjr, **AM** Amiga *General interest

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies:
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019
Address all inquiries to:
P.O. Box 5406, Greensboro, NC 27403

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. POSTMASTER: Send address changes to: **COMPUTE!** Magazine, P.O. Box 10955, Des Moines, IA 50950. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. ISSN 0194-357X

Editor's Notes

This month's Editor's Notes are written by Richard Mansfield, senior editor. We suggest that he does not mean to imply that "mouseketeers" are mousy; perhaps a rebuttal in the months ahead?

—Robert C. Lock, Editor in Chief

Ever since the Macintosh was introduced, the computing community has been debating about ease of learning versus ease of use: mice, menus, and icons are easy to learn, but typewriter keys, written commands, and control codes are often easier to use in the long run.

These two philosophies are represented rather neatly by two manufacturing giants, IBM and Apple. When you turn on an IBM, you are in the DOS environment. It's much like a programming language. There are dozens of words you can type which control the computer's behavior. Type DIR and you see a list of all the files on a disk. TIME will give you the time of day. CLS clears the screen. Beyond this, you can combine some of the commands: DIR > FILE sends a copy of the directory into a file named FILE. DIR | SORT will print a sorted directory. Essentially, you are given a rich language with which to communicate your particular instructions to your machine. But you pay a price for this richness—it takes longer to learn how to work with PC-DOS than it does to learn to use menu-driven systems like the Macintosh.

You may have seen the ads. A formidable tome crashes down next to a PC, graphically illustrating that running PC-DOS is a complicated affair. Then the Macintosh manual, light as a leaf, softly settles next to Apple's menu-driven computer. They're right, of course. You can be mousing around with the Macintosh within minutes, effortlessly deleting files, sorting directories, and activating applications programs.

Atari has chosen to configure its new ST computer quite like the Macintosh. The familiar elements are all in place. The ST displays icons (pictorial representations) so you can tell at a glance when something's a data file. It will look like a tablet with the edges of the pages turned up. On an IBM, by contrast, you must learn that filename extensions like .EXE or .COM signify a program that can be run. Extensions like .DOC indicate a data or text file.

On the IBM, you delete a file by typing DEL NAME. On the ST, it's a bit difficult to describe. You use the mouse controller to move a pointer on the screen to open a disk directory. Then you move the mouse to the target filename and click the mouse, highlighting the name. Then you click the mouse again and drag a picture of the filename until it's on top of a picture of a trash can. A warning window opens and asks you if you, in fact, *do* want to delete the file. You must either click the mouse in a box labeled CANCEL or in another box labeled OK. During this process, you must be able to *see* the filename and the trash can. Thus, if something is covered up, you must move it to some available space on the screen before you can access it. This can add steps to the above process. You might need to make some windows smaller or move them to a different part of the screen.

It sounds pretty intimidating, but skilled mouseketeers can fly around the screen, popping windows open and closed at quite a clip. You do need a fair amount of clear desk space to the side of the computer where you might otherwise have a book. But, one of the ideas behind windows, icons, and mice is that you won't need a book. Everything is on screen: windows covering older windows, menus popping out of other menus, "dialog" boxes appearing on top of menus. Your desktop is clean (for the mouse), but your screen can get pretty busy.

Although early STs are currently being shipped without software or documentation offering an alternative to the mouse environment, there is a command program which allows you to talk to the ST directly in the IBM style. In this mode, you can list a directory with the simple command LS. And you can quickly see everything in any data file via TYPE NAME. It's too early to tell whether or not this facility will be made part of the ultimate ST package. But that is the solution to the debate: offer both styles. For people who prefer not to type, offer mice. For people who don't like mice, offer command control. For people who prefer words like DEL, offer text-only screens. For people who prefer pictures, offer the trash can illustration.

Similarly, when you go to buy a word processor, one of the major factors in your decision will be whether you want a menu-driven or control-code-driven package. For example, some software pops up with a menu every time you want to change the margin: 1. Indent? 2. Flush right? 3. Single line? and you type the number signifying your choice. Additional menus might then appear asking how much you want to indent. Conversely, control-code style software requires that you memorize a pattern. To indent ten spaces, you might type CONTROL-I 10. This is a lot faster than responding to menus, but it is harder to learn and remember. If you indent often and are a good typist, however, you will likely prefer the efficiency of control codes. For one thing, your fingers don't leave the keyboard so commands to the word processor don't require that you look at the keys.

The best software offers the user a choice of either menus or control codes. Perhaps the best computers will offer optional mice, windows, and icons, but will provide a command-driven mode as well. When both styles are available, we can have the best of worlds.

Richard Mansfield

Senior Editor

Coming In Future Issues

All Machine Language Spreadsheet With *SpeedScript* Integrator For Commodore 64/128, Apple, Atari

Balloon Crazy: Frantic Arcade Game For Atari, Commodore 64, Apple, IBM PC/PCjr, TI

***SpeedScript* Revisited: Updates And Hints For Commodore 64, VIC, Atari, Apple**

Million-Color Palette For IBM PC & PCjr

Computed GOTOs & GOSUBs For Commodore 64

Atari Fine Scrolling

Apple Hi-Res Screen Dump

COMPUTE! Publications, Inc. publishes:

COMPUTE!
COMPUTE!'s GAZETTE
COMPUTE! Books
COMPUTE!'s GAZETTE DISK

Corporate office:
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408 USA

Mailing address: COMPUTE!
Post Office Box 5406
Greensboro, NC 27403 USA

Telephone: 919-275-9809

Subscription Orders

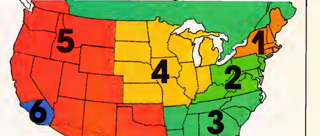
COMPUTE!
P.O. Box 10955
Des Moines, IA 50950

TOLL FREE Subscription Order Line
800-334-0868
In NC 919-275-9809

COMPUTE! Subscription Rates (12 Issue Year):

US	(one yr.)	$24
	(two yrs.)	$45
	(three yrs.)	$65
Canada and Foreign Surface Mail		$30
Foreign Air Delivery		$65

Advertising Sales



1. New England
Jonathan M. Just
Regional Manager
212-315-1665

2. Mid Atlantic
John Saval
Eastern Advertising Manager
212-315-1665
Kathy Hicks
Marsha A. Gittelman
215-646-5700
Brian S. Rogers
212-674-0238

3. Southeast & Foreign
Harry Blair
919-275-9809

4. Midwest
Gordon Benson
312-362-1821

5. Northwest/ Mountain/Texas
Phoebe Thompson
408-354-5553

6. Southwest
Ed Winchell
213-378-8361

Director of Advertising Sales
Ken Woodard

COMPUTE! Home Office 919-275-9809.

Address all advertising materials to:
Patti W. Stokes
Advertising Production Coordinator
COMPUTE! Magazine
324 West Wendover Avenue,
Greensboro, NC 27408

The COMPUTE! subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please send an exact copy of your subscription label to: COMPUTE! P.O. Box 10955, Des Moines, IA 50950. Include a note indicating your preference to receive only your subscription.

Readers Feedback

The Editors and Readers of COMPUTE!

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers' Feedback," COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.

Falling Through Trapdoors

I have a question about the placement of NEXT in a program. After typing in "Devastator" (COMPUTE!, August 1984) I made a few changes. In lines 1293–1294 (shown here) I tried moving the NEXT from line 1294 to the end of 1293. But now the program doesn't erase text the way it should. I thought it wouldn't make any difference which line the NEXT was on. Can you explain?

```
1293 FORT=1024TO1400:IF PEEK(T
     )<160THEN POKE T,32
1294 NEXT:GOSUB1300
```

Alfred Glasser, Jr.

The answer to your question applies to virtually every computer with BASIC. When the computer finds an IF statement, it immediately tests the expression after IF to determine whether it's true or false. If the expression is true, the computer performs whatever comes after THEN on that line. If the expression is false, the computer ignores everything after THEN and goes directly to the next program line. When an IF test proves false, it's as though a trapdoor opens at THEN. The computer immediately falls through *(proceeds) to the next program line and performs what it finds there.*

The lines shown here test screen memory locations 1024–1400. In plain English the part before GOSUB 1300 means "Check every location from 1024 to 1400. If a location doesn't contain a reverse space character (160), replace it with a blank (32). Otherwise ignore it." If the expression PEEK(T)<160 is true, the computer executes POKE T,32 before going to NEXT in line 1294. If the expression is false—if the location contains a reverse space—the computer skips the part after THEN and immediately falls through to 1294. Note that NEXT is always performed whether the IF statement is true or false. Moving NEXT to the end of 1293 causes it to be executed only when the IF test is true—clearly not what the programmer intended.

Because the computer falls through an IF-THEN statement when the test proves false, be careful what you add to IF lines. Don't add statements to the end of the line unless you want them to be performed only when the IF test is true. For similar reasons you shouldn't put anything on the same line after a GOTO statement (which immediately sends the computer somewhere else in the program). These two lines demonstrate the error: The GOTO in line 10 prevents NEVER from being printed.

```
10 GOTO 20:PRINT "NEVER"
20 PRINT "ALWAYS"
```

Atari Disk Speedup

I have a solution for Duyen Nguyen, who asked for a way to speed up his Atari disk drive ("Readers' Feedback," July 1985). Enter POKE 1913,80 to disable the verify function. Your drive will run faster.

Jim Noland

Thanks for pointing this out. This POKE dramatically speeds up write operations and has been widely used by Atari owners for years. In fact, some Disk Operating Systems, such as OS/A+ and DOS XL, incorporate this modification by default. The POKE works by modifying DOS to turn off the write-with-verify function. Normally, location 1913 contains the value 87, which tells DOS to verify each sector as it is written. This assures an error-free SAVE but also slows things down considerably. Disabling this function with POKE 1913,80 can make a noticeable difference. Although you might expect the modification to increase the likelihood of errors, in practice this is extremely rare. Atari programmers at COMPUTE! have been using this technique for many years without problems.

To save yourself the trouble of performing this POKE each time you boot your system, you can save the modified DOS on disk. After entering the POKE, type DOS. When the DOS menu appears, select option H, "Write DOS Files."

The new Atari DOS 2.5 disables write-with-verify by default. It also lets you change this function without making any POKEs. Simply run the DOS 2.5 utility file SETUP.COM and select the option "Change System Configuration." This is safer than POKEing around in DOS, because a mistyped POKE command could mess up something.

ProDOS Date And Time

I have numerous books covering my Apple IIc and the ProDOS operating system, but nowhere have I been able to find out how to set the ProDOS date and time. Can you help me with this?

Stanley Moody

ProDOS keeps information about the current date and time in its System Global Page, a 256-byte block of memory starting at location 48896 ($BF00). On an Apple IIe this information can be updated by a clock card. The Apple IIe User's Disk also has a utility to let you set these locations. The following program permits you to set date and time on the IIc.

```
99 10 REM SET TIME AND DATE
95 20 PRINT "TODAY'S DATE (MM/DD
      /YY) ->";: INPUT D$
8C 30 IF LEN (D$) < > 8 THEN GOS
      UB 1000: GOTO 20
2A 40 Y = VAL ( MID$ (D$,7)) * 2
      :M = VAL ( MID$ (D$,1,2)):
       IF M > 12 THEN GOSUB 1000
      : GOTO 20
6B 50 IF M > 7 THEN Y = Y + 1:M
      = M - 8
B3 55 D = VAL ( MID$ (D$,4,2)):
      IF D > 31 THEN GOSUB 1000:
       GOTO 20
BC 60 D = D + M * 32
92 70 POKE 49041,Y: POKE 49040,D
19 80 PRINT "TIME TO STAMP ON FI
      LES (HH/MM) ->";: INPUT T$
C9 90 IF LEN (T$) < > 5 THEN GOS
      UB 1010: GOTO 80
4D 100 H = VAL ( MID$ (T$,1,2)):
        IF H > 24 THEN GOSUB 101
       0: GOTO 80
EF 110 M = VAL ( MID$ (T$,4,2)):
        IF M > 59 THEN GOSUB 101
       0: GOTO 80
CD 120 POKE 49043,H: POKE 49042,
       M
8F 130 END
CD 1000 PRINT "BAD FORMAT FOR DA
       TE": RETURN
D2 1010 PRINT "BAD FORMAT FOR TI
       ME": RETURN
```

You've Stumbled Into An Unknown Computer System.

Now What?

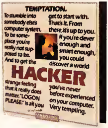
Welcome To The Next Generation Of Home Computer Software.

"Plunge into every hacker's dream."
—Rick Manning/Louisville Times

"...the most mysterious game ever."
—Rawson Stovall/"The Vid Kid"

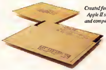
"An open screen says 'logon please.' That's it, no instructions."
—Technology Notebook/Gannett News Service

Hacker™



"Just like in real life, there are no rules, no clues, no instructions."
—David Greising/Technology Memo

What more can we say? Well, someone as clever and smart as you certainly wouldn't want any hints, right? So all we'll say is it was created by legendary designer, Steve Cartwright.



Created for the Commodore 64/128, Apple II series and Atari 800, XE, XL and compatible systems.

ACTIVISION
HOME COMPUTER SOFTWARE

The Next Generation.

You can find Mindscape's The Halley Project at these fine stores:

National
Available at all Child World and Childrens Palace Stores
Federated Group
Sears — Most larger stores
Waldenbooks — Most locations

Alabama
Software City
Huntsville
The Computer Shoppe
Huntsville

Alaska
Computer Express
Fairbanks

Arkansas
The Computer Shoppe
Little Rock

California
Access to Software
San Francisco
Affordable Computer Systems
Santa Clara and San Jose
Alamo Computer Center
Cupertino
Alamo Electronic Components
San Jose
Applied Computer Skills
Van Nuys
B. Dalton Software Etc
Concord, San Diego, Bakersfield, Torrance, Montclair, Northridge
Books & Bytes
Cypress
Boot's Camera Electronics
Fresno
Border Software
Chulavista
Brown Knows Computer
Redlands
Candy Computer
Elk Grove
Computer Outlet
San Diego
Computermart
Diamond Bar
Computertime
Citrus Hts
Corporate Micro Services
San Francisco
Crown Book & Software
Los Angeles, Santa Monica, Marina Del Rey, Studio City, Santa Ana, Redondo Beach, Hollywood, Glendale, Culver City, South Pasadena, Thousand Oaks, Torrance, Huntington Beach, Downey, Sherman Oaks, Encino, El Toro, Long Beach, Palos Verdes, Northbridge, Pasadena, Westminster, Woodland Hills, Ventura, Lakewood
Egghead Software
Lawndale, Pasadena, Orange, Sherman Oaks, San Diego
Family Discount Computer
Escondido
Future Vision
San Louis Obispo
Futurvision
Napa
HT Electronic, Inc
Sunnyvale
Home Computing Centers
San Bruno, San Leandro
Learning Tree Computer
Santa Ana
Software Centre Int'l
Costa Mesa, Santa Ana
Software Shoppe
Huntington Beach
Software First
San Rafael and Santa Rosa
Software Galeria
Orange
Software House
Fresno
Software Service Center
Tustin
Software Solution
Chico
Software Solutions
El Cerrito
Software Station
Costa Mesa
Software Supermarket
Lawndale, Los Angeles
Sherman Oaks
Software World
Redding
Software, Etc
Novato
Sysware
San Diego

Colorado
Citadel Computer
Colorado Springs
Colborn School Supply Co
Denver
Colorado Computer Systems
Westminster

Colorado (cont)
DMA Computer Solutions
Colorado Springs
Kazoo & Company
Denver
Software Center
Concord

Connecticut
Caldor
West Norwalk, Old Saybrook, Enfield, Milford, W Hartford, Vernon, Riverside, Norwalk, Brookfield, Trumbull, Mansfield, Norwich, Hamden, Waterbury, Manchester, Stamford, Wallingford, Avon, Rocky Hill, Ridgefield, Torrington, Branford, Bristol, Meddletown, Fairfield, Groton, New Britain, Derby, Southington
Crazy Eddie
Norwalk
Danbury Video & Computer
Danbury
Derric Electronics
Hamden
Kiddytown
Norwalk
Personal Computer Center
Norwich
Petzold's
Seymour
Softown, Inc
Danbury
Software City
Stamford, Orange
Software Kingdom
E Hartford, E Windsor
Video Connection
Bridgeport

Delaware
Software Kingdom
Dover

District of Columbia
Crown Book & Software
(3 locations)

Florida
Computer Chip
Sarasota
Computer Generation
Plantation
Computers, Etc.
Lakeland
Discount Disks
Shalimar
Florida Micro Media
Pompano Beach
Games 'N Gadgets
Clearwater, Jacksonville, Tallahassee, Orange Park
Maxicat
Miami
Program Store
Clearwater, Pinellas Park
Software Cellar
Fort Lauderdale
Software Shops
Brandon

Georgia
Academic Software
Roswell
Alpha Information Systems
Fayetteville
Cardinal Computers
Dalton
Electronics Boutique
Atlanta — Lenox Square
Cumberland Mall
Games 'N Gadgets
Atlanta — North Lake Mall, Perimeter Mall
Software Atlanta
Atlanta (2 locations)
Software City
Atlanta
Versatile Video
Atlanta and Roswell

Hawaii
Software Library
Honolulu

Idaho
Software Galeria
Idaho Falls
Stokes Brothers, Inc
Boise

Illinois
ABC Diskette
Chicago
Asoftware Store
Oaklawn
Apprat Computers
Arlington Heights
B Dalton Software Etc
Chicago
C02 Enterprises
Chicago
Complete Computing
Lombard
Computer Outpost
Schaumburg

Illinois (cont)
ComputerLand
Niles
Computerworld
Chicago
Crown Book & Software
Orland Park, Niles, Wheaton, Evanston, Oaklawn, Oak Park, Chicago, Skokie, Deerfield, Downers Grove, Highland Park, Naperville, Hoffman Estates, Arlington Heights, Calumet City
Disk-N-Tech
Orland Park
FD Software & CADS
Roselle
Family Software
Glendale Heights
Software and Beyond
Schaumburg
Software City
Arlington Heights
Software Plus West
Hanover Park
Susie Software
Mount Prospect

Indiana
B Dalton Software Etc
Indianapolis
Burkat Computer Center
South Bend
Computer Corner
Merrillville
Microcomputer, Inc
Indianapolis

Kentucky
Software Source
Louisville
The Computer Shoppe
Louisville (2 locations)
Videovisions
Louisville

Louisiana
B Dalton Software Etc
New Orleans
PFL Computer Center
Monroe
Software Mart
Metarie

Maryland
Buried Treasure
Rockville
Caldor
Timonium, Baltimore, Severna Park, Ellicott City
Crown Book & Software
Wheaton, Columbia, Laurel, Kensington, Silver Springs, Bowie, Bethesda, Greenbelt, Gaithersburg, Annapolis, Rockville
Electronics Boutique
Baltimore
Games 'N Gadgets
Columbia, Baltimore (White Marsh Mall, Security Square Mall, Eastern Ave.)
Greetings and Readings
Towson

Massachusetts
Atlantic Computers
Waltham
Barnes & Noble Bookstore
Boston
Caldor
Chelmsford, Westborough, Swansea, Pembroke, Westfield, Auburn, Malden, Norwell, Brockton, Canton, Brighton, Taunton, Salem, Chicopee, Stoneham, Springfield, Framingham, Northampton, Worcester, Weymouth, Saugus, Methuen, Burlington
Computer Concepts
Hanover
Computer Etc
Peabody
Electronic Boutique
Holyoke
Softpro
Burlington
Software City
West Springfield
Software Galeria
Boston
The Whiz
Westboro

Michigan
Micro World
Livonia and Milford
VorTech Computer Center
Pontiac

Minnesota
B Dalton Software Etc
Brooklyn Park, Minnetonka, Roseville, Edina, Minneapolis

Missouri
Forsyth Computers
St Louis

Missouri (cont)
Software To Go
St Louis and Clayton

Montana
Colborn School Supply Co
Billings

Nevada
Century 23
Las Vegas
Computer House
Sparks
Computer Video Electronic
Carson City
Software City
Las Vegas

New Hampshire
Caldor
Bedford
Micon Computer
Londonderry
Neba Computer
Plaistow
Portsmouth Computer Center
Portsmouth
Soft Spot
Manchester

New Jersey
Bamberger's
Menlo Park, Cherry Hill, Paramus, Livingston, Wayne
Caldor
Eatontown, S Plainfield, West Orange, Toms River, Watchung, Paramus, Morris Plain, Woodbridge, Brick Town, N Brunswick, W Paterson
CIA
S Plainfield
Circuit Services
Newton
Computer Pro's
Wayne
Computer Software
Ocean
Crazy Eddie
Paramus, E Brunswick, Union, Totowa, Woodbridge
Disk-of-the-Month Club
Fair Lawn
Electronics Boutique
Voorhees, Rockaway, Woodbridge
Family Computer Centres
Fairfield, South Orange
Games N Gadgets
Livingston, Burlington, Wayne
Gemini Enterprises
Cedar Knolls
Golden Hedge Inc
Cherry Hill
Igor Software
Teaneck
Kearny Electronics
Kearny
Prince Range
Livingston
Program Store
Eatontown and Wayne
Softec
Newton
Software City
Redbank, Linwood, Ridgefield, Summit, Teaneck, Bergenfield, Parsippany, Linwood
Software Spectrum
North Plainfield
Software Station
Morristown and Rockaway
Video Dynamics
Ridgewood
Videomania/Computers Plus
Bloomfield
Village Computer Center
Cedar Knolls
Wayne Software
Wayne
Wolsten's Inc
East Orange

New Mexico
Academy Computer
Albuquerque
B Dalton Software Etc
Albuquerque
Colborn School Supply Co
Albuquerque

New York
17 Software Street
Vailsgate
45th Electro
New York
47th St Photo Inc
New York (3 locations)
Advanced Camera
New York
Annex Outlet
New York
Barnes & Noble Bookstore
New York (2 locations)

New York (cont)
B Dalton Software Etc.
Yonkers and New York
Big L Electronics
W Hempstead, W Babylon, Syosset, New Hyde Park, Patchogue
Binary Orchard Inc
Hamburg, West Seneca
Broadway Computers
New York
Byte Shop
Merrick
Caldor
Lake Ronkonk, East Patchogue, Wappingers Falls, Riverhead, Rocky Point, Bridgehampton, Vails Gate, Glens Falls, Kingston, Middletown, Latham, Mahopac, Bedford Hills, Albany, Yorktown Heights, Peekskill, Pelham Manor, Schenectady, Nanuet, White Plains, Yonkers, Newburgh, Port Chester, Poughkeepsie, W Babylon, Levittown, Coram, Clifton Park
CIA Software
Flushing
Computer Discount of America
Huntington
Computer Software Plus
Brooklyn
Computer Tech
Brooklyn
Computerware
East Meadow
Computerworld
Bayshore
Crazy Eddie
Nesconset, Mass, New York, Syosset, Elmhurst, Brooklyn, Carle Place, Hartsdale, Bronx, Nanuet
East 33rd Typewriter
New York
Electronic Man
New York
Electronics Boutique
Albany
Executive Photo & Supply
New York
Focus Electronic
Brooklyn (2 locations)
Four Guys
New York
Games N Gadgets
Garden City, Huntington, Nanuet
Glossy Photo
New York
Great American Software
Flushing
Home Software
Albany
Intercontinental
Flushing
Island Video & Computer
Coram
J&R Computer Outlet
New York
J&S Electronic
New York
Leigh's Computers, Inc
New York
Lloyd's
New York
Lloyd Corner
New York
London Luggage
New York
Micro Co
New York
Micro Electronics
Valley Stream
Montgomery Grant
New York
Park Avenue Video
New York
Photo Sound
New York
Programs Plus
Brentwood
Quartz Electronics
New York
Quo Vadis
Ridgewood
Show & Tell Video
Bayshore
Software Centre
Forest Hills
Software City
Mount Kisco, Syracuse, Wappinger Falls, Brooklyn, Flushing, Tonawanda, Spring Valley
Software Link
White Plains
Software Seller
Harrison
Staten Island Software
Staten Island
Sysutil Inc
Brooklyn
Thruway Market
Walden

New York (cont)
Wall Street Camera
New York
Willoughby's
New York
World Trade
Bayside

North Carolina
Byte Shop
Greensboro
Games N Gadgets
Raleigh
Oasis Software
Wilmington

North Dakota
Colborn School Supply Co
Grand Forks
Software House
Fargo

Ohio
Abby's Software
Fairborn
B Dalton Software Etc
Columbus
Big Bytes Computers
Poland
Computers Plus of Ohio
Mansfield
Disk Drive
Toledo
Electronic Connexion
Kettering
Electronic One
Columbus
Games N Gadgets
N Randall
Magic One Computer
Barberton
Microwave Magic
Fairfield
North Coast Programming
Willoughby and Solon
Program Store
Columbus
Software and More
Cincinnati
Software City
Columbus
Swallens
Cincinnati
Sylvan Enterprises
Cincinnati
Video Computer World
Oregon
Video HD
Cleveland

Oklahoma
Computer Software Center
Oklahoma City
Home Software
Tulsa and Oklahoma City
Videocomp
Lawton

Oregon
At Computers
Klamath Falls
Egghead Software
Beaverton
Link Three Inc
Beaverton
Software Galeria
Beaverton

Pennsylvania
B Dalton Software
Monroeville
Bamberger's
Whitehall, King of Prussia
Black Sun
Emmause
De Re Computers
Harrisburg
Electronics Boutique
Langhorne, King of Prussia, Whitehall, Exton
Floppy Disk
Mechanicsburg
Games N Gadgets
Lancaster, West Mifflin, King of Prussia, Media, Philadelphia

Rhode Island
Caldor
Woonsocket, Warwick
Lincoln
Future Images
Woonsocket
Software Connections
Warwick

South Carolina
Horizon Commodore Super Store
Greenville (2 locations)
Software Solutions
Charleston

Tennessee
Games N Gadgets
Memphis

Tennessee
Inacomp
Brentwood
Software Store
Memphis
The Computer Shoppe
Nashville, Knoxville, Memphis, Chattanooga, Madison

Texas
B Dalton Software Etc
Houston
Babbages
Dallas, Houston, Plano, Mesquite, Fort Worth, Irving, Hurst
Casa Computer
Lubbock
Compurite
Houston (2 locations)
Computer City
San Antonio
Computer Experience
San Antonio
The Computer Store
San Angelo
Data Pro Computer Center
Wichita Falls
Edu Tron
Fort Worth
Floppy Wizard
Houston
Frisco Electronics
San Antonio, Austin
Home Software
Friendswood
National Video Austin
Austin (4 locations)
Software & Things
Austin
Software Store
San Antonio
Videoland
All locations

Utah
Software Center
Salt Lake City
Software Hut
Salt Lake City
Stokes Brothers, Inc
Salt Lake City, West Valley, Orem, Ogden, Layton

Virginia
Computer Works
Petersburg
Crown Book & Software
Alexandria, Woodbridge, Reston, Springfield, Vienna, Manassas, Mclean, Falls Church, Fairfax
Family Computer Center
Fairfax
Games 'N Gadgets
Hampton, Virginia Beach
L&Y Electronics
Woodbridge
Next Generation Computer
Williamsburg
Program Store
Falls Church
Software Central
Norfolk
Software Mart
Springfield
Softwarecenter Inc.
Vienna
University Bookstore
Blacksburg

Washington
Butler's TV and Computer Service
Federal Way
Egghead Software
Tukwila, Bellevue
Nibbles and Bites
Tacoma

Wisconsin
Computer Software Center
Milwaukee
Computer Warehouse
Milwaukee
Romeyn Software
Janesville
Software Source
Brookfield
TMW Software
Wausau

Wyoming
Colborns of Wyoming
Casper

Canada
Compusoftware
Vancouver, BC

Note: Not all formats available at all locations

If no dealer in your area, phone 1-800-221-9884 for information. (Illinois - 1-800-942-7315)

MINDSCAPE

Hi-Res Characters On The 64

I have written a program that draws charts and graphs on the Commodore 64's high-resolution screen, but have trouble putting numbers and letters on the screen. Plotting every character pixel by pixel takes much too long. Is there any easy way to do this?

Sean Wood

One solution is to copy the character definitions directly from the ROM (Read Only Memory) character set into the bitmap. The following program demonstrates the technique. Lines 10–30 enter hi-res mode, lines 100–180 contain the character plotting routine, and line 40 shows how to call the routine. Define the message you want to print as A$. Variables X and Y determine the row and column where printing begins. Keep X within the range 0–39 and Y in the range 0–24. DX controls the direction of printing. If DX=1, the string prints from left to right; if DX=40, it prints from top to bottom. Other values can be used to print diagonally, from bottom to top, and so on. BK and CH set the background color and character color, respectively. After these variables are defined, GOSUB 100 puts the string on the screen.

Another solution is to look up the article "64 Multicolor Graphics Made Easy" in the October issue of COMPUTE!. *It includes a program called "Color Plotter 64" that adds 14 commands to Commodore BASIC for drawing multicolor hi-res graphics and text.*

```
10 POKE53265,PEEK(53265)OR32
20 POKE 53272,PEEK(53272)OR8:P
   RINT"{CLR}"
30 BASE=8192:FORA=BASETOBASE+8
   192:POKEA,0:NEXT:REM CLEAR
   {SPACE}HIRES SCREEN
40 A$="ABCDEFGHIJKLMNOPQRSTUVW
   XYZ1234567890":X=0:Y=0:DX=1
   :BK=1:CH=6:GOSUB100
50 WAIT198,1:POKE53272,21:POKE
   53265,27:PRINT"{CLR}":END
100 S=X*8+Y*320+DASE:D=1024+X+
    40*Y
110 FOR A=1 TO LEN(A$):B=ASC(M
    ID$(A$,A,1))
120 IF B>63 AND B<96 THEN B=B-
    64:GOTO 140
130 IF B>95 THEN B=B-32
140 C=B*8+53248:POKE56334,0:PO
    KE1,51:POKED,BK+16*CH
150 FORQ=0TO7:POKES+Q,PEEK(C+Q
    ):NEXT
160 POKE1,55:POKE56334,1
170 S=S+DX*8:D=D+DX:NEXT
180 RETURN
```

Commodore Screen Splitting

Is there any way to split the Commodore 64's screen between multicolor bitmapping on the top and uppercase text on the bottom?

Brian Sullivan

The picture on your TV or monitor is composed of many horizontal lines called raster lines. *The 64 permits you to set up an interrupt at any raster line. When the computer reaches that line, it stops what it's doing and performs a special machine language routine (which you must have prepared in advance). This technique, known as raster interrupt programming, is covered thoroughly in* COMPUTE!'s First Book Of Commodore 64 *and* Mapping The 64. *Here's a program that puts a multicolor bitmap display at the top of the screen and uppercase text at the bottom. POKE location 2 with the number of the raster line where you want the change to occur (only lines 50–249 are visible on the screen).*

```
10 FORA=828TO913:READB:POKEA,B
   :C=C+B:NEXT:IFC<>9673THENPR
   INT"{CLR}DATA ERROR":STOP
15 SYS828
20 DATA 120,169,BB,141,20,3,16
   9,3,141,21
30 DATA 3,169,1,141,26,208,169
   ,27,141,17
40 DATA 208,88,169,127,141,13,
   220,96,169,1
50 DATA 141,25,208,162,59,160,
   216,173,18,208
60 DATA 197,2,176,9,169,29,141
   ,24,208,165
70 DATA 2,208,11,162,27,160,20
   0,169,21,141
80 DATA 24,208,169,0,142,17,20
   8,140,22,208
90 DATA 141,18,208,173,13,220,
   41,1,240,3
100 DATA 76,49,234,76,188,254
```

Commodore Countdown

I am writing a Commodore program and want to add a timer that counts down in minutes and seconds. My problem is that when the timer reaches 0 it flips to 99 instead of 59. Can you help?

Chaiyos Gosolsatit

*In many cases it's easiest to treat time as seconds rather than minutes and seconds. Then you have only one number to worry about. When you need to display the time, convert the number of seconds into appropriate minute and second values. For instance, if TM represents the number of seconds, the statements MN=INT (TM/60) and SE=TM−60*INT (TM/60) calculate the minutes and seconds, respectively.*

The following routine demonstrates a simple countdown timer that should work on any Commodore computer. Line 10 sets the computer's internal clock to 000000. The reserved variable TI$ returns the time (in hours/minutes/seconds format) elapsed since reset. As shown, the example provides a countdown of three minutes (180 seconds). To modify this, change the value of SS (line 10) to the desired number of seconds.

```
10 TI$="000000":SS=180
20 T$=TI$:TM=SS-(VAL(MID$(T$,3
   ,2))*60+VAL(MID$(T$,5,2)))
30 MN=INT(TM/60):SE=TM-MN*60
40 PRINT"{HOME}"MN"{LEFT}"SE"
   {LEFT} ":GOTO20
```

Atari Cartridge Dilemma Solved

Like many other Atari owners, after suffering from the bugs in revision B BASIC, I ordered the new revision C BASIC cartridge for my 800XL. However, with the BASIC cartridge in place I can't use the *Monkey Wrench II* cartridge (a useful BASIC editing aid). My solution is this program, which copies the old BASIC from ROM into underlying RAM with a fast machine language routine, then changes rev B into rev C (only 12 bytes are different). This program runs so fast that it's almost as convenient as plugging in a cartridge, and now I can use my editing cartridge along with the new BASIC. Pressing RESET switches ROM BASIC back in; enter POKE 54017,255 to go back to rev C BASIC in RAM.

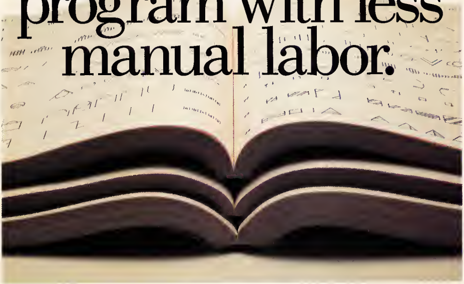
```
1 FOR I=0 TO 43:READ A:PO
  KE 16384+I,A:NEXT I:A=U
  SR(16384)
2 DATA 104,169,0,133,203,
  169,160,133,204,162,32,
  160,0,177,203,72
3 DATA 169,255,141,1,211,
  104,145,203,169,253,141
  ,1,211,136,208,237
4 DATA 230,204,202,48,6,2
  08,230,160,0,208,226,96
5 FOR J=1 TO 13:READ A,B:
  POKE A,B:NEXT J
6 DATA 54017,255,43231,23
  4,43232,240,43233,17
7 DATA 43234,234,47913,0,
  49139,0,49140,0,49141,0
8 DATA 49142,0,49143,0,49
  144,0,49145,0
9 PRINT "BASIC VERSION C
  ACTIVATED":PRINT "POKE
  54017,255 TO REACTIVATE
  "
```

Gregory Latta

Thanks for the program, which should prove useful to Atari owners who wish to use other cartridges with the new BASIC. The revision B bugs, found in the BASIC built into the 600XL and 800XL, are familiar to many Atari users by now. See Bill Wilkinson's "INSIGHT: Atari" column in June 1985 COMPUTE! *for a demonstration of the bug that mangles strings. To demonstrate the bug that adds 16 bytes to a program when you load it, run the program above, then enter POKE 54017,253 (or press RESET) to switch the ROM BASIC back in. Now type in and run the following program (a disk drive is required):*

```
1 ? "PROGRAM ENDS AT ";PE
  EK(140)+256*PEEK(141):?
   "# OF BYTES FREE ";FRE
  (0)
```

Choose the SAT program with less manual labor.



The idea behind choosing a computerized SAT program over a manual is to save you from piles of paperwork. But surprisingly, two of the best-known programs come with big, fat manuals and only 2 or 3 double-sided disks.

When that much information is put into the manual, what's left to put into the computer?

Why not buy a computer program that's really a computer program? Buy The Perfect Score from Mindscape for just $69.95.*

It has 6 double-sided disks and a real skinny manual. It even has printout capability and a continuous on-screen clock. All this makes The Perfect Score more computerized than those others.

Now, if this cold logic fails to convince you, perhaps an emotional appeal to your sense of patriotism and social consciousness will. Your choice is this. Either you buy their SAT, which kills innocent trees to make all that paper. Or you buy our SAT with 6 disks and Save America's Trees.

The Perfect Score, $69.95, from Mindscape

```
2 SAVE "D:EXPANDER":IF PE
  EK(53279)<>6 THEN RUN "
  D:EXPANDER"
```

The program saves, reloads and runs itself over and over, growing 16 bytes longer every time when rev B BASIC is present. Press the START key when you've seen enough. Now enter POKE 54017,255 (to switch in rev C BASIC), then run it again to confirm that it saves and reloads without changing in size.

Atari ML Addresses

I own an Atari 800XL and was interested in the "Commodore ML Addresses" program in "Readers' Feedback," September 1985. Do you have a program for Atari computers that finds the starting and ending addresses of machine language programs on disk and tape?

Adam Mercadante

This program prints the starting and ending addresses of most machine language files. Be sure to include the C: prefix (for tape) or D: prefix (for disk) when entering the filename.

```
10 DIM A$(14)
20 PRINT "ENTER FILENAME
   (INCLUDE C: OR D:)":IN
   PUT A$
30 OPEN #1,4,0,A$:GET #1,
   A:GET #1,A
40 GET #1,SLB:GET #1,SHB
50 GET #1,ELB:GET #1,EHB
60 PRINT "START ADDRESS =
   ";SLB+256*SHB
70 PRINT "   END ADDRESS =
   ";ELB+256*EHB
80 CLOSE #1
```

IBM Compatible Coverage

Now that the PCjr has died, I begin to worry anew about what little support and information has been forthcoming for the IBM-compatible home computers. (I define that as an MS-DOS-based 8088 chip computer which can be purchased for less than an Apple IIe system.) So far I have been able to run all the PCjr programs in COMPUTE! on my Tandy 1000. And all the programs in your book *Easy BASIC Programs for the IBM PC and PCjr* run beautifully on my Tandy. I recently bought your machine language book for the PCjr and have not run into problems yet. But now I fear for the future of those books; you might be tempted to pull them off the shelves before they even become available. Please don't. I appeal to your business sense to broaden the spectrum of your coverage and pay some attention to the market so strikingly similar to the IBM market you already cover. Why not change your PCjr coverage into PC/MS-DOS coverage? This surely requires only a minimum of effort and I think it will pay off.

Christopher L. Herd

Our home-oriented IBM coverage in COMPUTE! *already is directed toward compatibles as well as both the PC and PCjr. If your "IBM-compatible" computer is truly compatible, it should run the programs we publish for the PC and PCjr without modifications—as your experience with the Tandy 1000 bears out. The Tandy has proven to be highly compatible with IBM computers. But not all so-called compatibles are created equal. If a program doesn't run, there's almost certainly a slight compatibility problem with your computer, BASIC, or DOS. Since there are dozens of IBM compatibles on the market, it isn't practical for us to test every program on every system. Instead, we design the programs to work on what is considered the common denominator in the IBM-compatible world—the IBM PC itself.*

Commodore ML Keyboard Input

I'm writing a Commodore 64 machine language program that requires input from the keyboard to be printed on the screen. Neither the CHRIN routine ($FFCF) nor GETIN ($FFE4) seem to work properly, and after several weeks of work I'm stumped. The bug in question occurs only when I call the CHROUT routine with JSR $FFD2. When I JSR to $F1CA (the address $FFD2 jumps to), my program works fine. What's the difference between calling CHROUT at $F1CA instead of $FFD2?

Jerry Ford

Under normal circumstances it makes no difference which address you use. Since the Kernal call at $FFD2 simply performs JMP ($0326) to get to $F1CA, the result is the same unless you've disturbed the vector at $0326–0327. We can't debug your program without seeing the code, but you should know that CHRIN and GETIN handle keyboard input quite differently. Here are two brief examples that do the job you describe and show how the two routines differ. You'll need a machine language assembler to type them in (the comments are optional).

```
      LDX #0        ;Set counter
      STX TEMP      ;at zero.
LINE  JSR $FFCF     ;Input line/char.
      CMP #13       ;RETURN
                    character
      BEQ EXIT      ;terminates.
STORE LDX TEMP      ;Get counter.
      STA BUFFER,X  ;Store char.
      INC TEMP      ;Bump counter.
      BNE LINE      ;Always branch.
EXIT  RTS
TEMP  .BYTE 0
BUFFER = *
```

This routine puts the input string in memory starting at BUFFER and records its length in the variable TEMP. The code may look confusing unless you understand that CHRIN performs two different functions depending on when it's called. The first time you call CHRIN, the computer simply lets you enter a logical line (up to two screen lines). It displays a blinking cursor and allows you to type on the screen, waiting until you press RETURN. When CHRIN terminates, the accumulator holds the first character from the input line. At this point, the routine falls through to STORE to put the first character in BUFFER. BNE LINE goes back to do another JSR $FFCF, but this time CHRIN doesn't input a line. Instead it puts the second character in the accumulator. Subsequent calls to CHRIN retrieve the remaining characters, so the routine keeps storing and branching back until a carriage return appears. Calling CHRIN after the whole input line has been retrieved starts the process over again.

```
      LDX #0
      STX TEMP
GETIT JSR $FFE4    ;Get character.
      BEQ GETIT    ;Ignore nulls.
      CMP #13
      BEQ EXIT
      JSR $FFD2
      LDX TEMP
      STA BUF-
      FER,X
      INC TEMP
      BNE GETIT
EXIT  RTS
TEMP  .BYTE 0
BUFFER = *
```

GETIN does nothing but pull a character from the keyboard buffer and return it in the accumulator. Thus, if you want a cursor or editing keys, your program must provide them (we don't have space for a complete example here).

At first, CHRIN seems more useful than GETIN because it provides so many features (cursor, editing keys, etc.) automatically. But you pay a price for all that convenience. The first call to CHRIN traps you in the ROM routine until RETURN is pressed. If you type only what the program expects, all is well. But there's nothing to prevent a user from moving the cursor to the wrong line, clearing or scrolling the screen, typing graphics garbage rather than letters, or wreaking other sorts of havoc. To avoid such problems, it's often preferable to write a custom input routine with GETIN, adding code to handle editing keys, screening out unwanted characters, and displaying a cursor. The commented source code in SpeedScript: The Word Processor for the Commodore 64 and VIC-20 *(published by* COMPUTE! *Books) includes two fairly elaborate keyboard routines built around GETIN.* ©

Trends in Telecomputing

If you're a telecomputing enthusiast, how would you like to dial all the long-distance calls you want for only a modest monthly fee? Or access an online information service with color graphics for pennies a day? These and similar experiments may soon boost personal telecomputing to new heights of popularity.

Selby Bateman, Features Editor

Many companies are betting that telecommunications holds the key to the future of personal computing. Some of these companies are now experimenting with innovative ideas and lower prices.

For example, when you log onto an electronic bulletin board or online information service, minutes have traditionally been measured in dollars and cents. In effect, a meter is running for every moment you spend on the long-distance telephone line or carrier systems such as Telenet, Tymnet, and Uninet.

But now one of those carriers, GTE Telenet, is experimenting in a dozen major cities with a system that could drastically change the telecomputing landscape. For the first time, people in those cities will be able to call bulletin boards, other computer users, and noncommercial databases over the Telenet system for a flat monthly fee of $25. Without flat-rate billing, many telecomputing fans can amass $25 in charges in just one evening. The new service is called PC Pursuit.

There are limits to this experiment, however. PC Pursuit is available only during evenings and weekends, and cannot be used to access the commercial online services which have direct links with Telenet, such as CompuServe, The Source, Dow Jones, and others. Those systems have their own hourly rates which include access through Telenet and other long-distance carriers.

Still, PC Pursuit is a significant development for those who frequently call local bulletin boards and fellow computerists. The experiment is now under way in Atlanta, Boston, Chicago, Dallas, Denver, Detroit, Houston, Los Angeles, New York, Philadelphia, San Francisco, and Washington, D.C. Whether or not PC Pursuit expands into a national service depends on how much interest is generated.

Measuring Demand

"We've seen this as a need, but whether the potential market is great enough, we weren't sure—we still aren't totally sure," says Claudia Houston, a GTE Telenet spokesperson. "We're the first ones to have done this, so there's no proof."

Telenet's primary business is not the evening and weekend access which it makes available to consumers, Houston says. "The reason we're able to offer a rate like this is because we have the Telenet data network in place, a major value-added network service supplying business customers during the day. We're able to handle a billion packets of data a month, equivalent to about 28 million typed pages. So when business closes up at the end of the day, there's plenty of room for other uses."

To use PC Pursuit, you first call the local Telenet number, then enter your name and phone number. Next you enter the name of the city you're calling and the phone number, then hang up. PC Pursuit makes the contacts and calls you back with the connection already established. The service prevents illegal use of the long-distance network for voice connection. Each month, PC Pursuit customers are billed automatically on their Visa or MasterCard accounts.

GTE Telenet is eager to hear

from people who are interested in PC Pursuit, even if you don't live in one of the 12 cities involved in the experiment. A toll-free bulletin board has been set up to distribute more information, and you can also leave a private message about PC Pursuit for Telenet's ongoing market research. The bulletin board number is 1-800-835-3001. For voice phone inquiries between 8 a.m. and 5 p.m. Eastern time, call 1-800-368-4215.

If PC Pursuit catches on, it can be easily extended to other metropolitan areas, Houston adds. In one form or another, the idea behind PC Pursuit will eventually be established, agree observers: easier, cheaper access for nonbusiness personal telecomputing.

The Quantum Connection

People who use computers at home are beginning to wake up to the possibilities of telecommunications, says Owen Davies, co-editor of *The Omni Online Database Directory,* an annual compendium of more than a thousand electronic databases. Business people may now make up the bulk of the traffic, but individuals are finding new applications almost every day. Davies, who closely watches the telecomputing field, has seen plenty of growth during the past year: new online databases in many different areas of interest, easier access for home users, and telecomputing software that's simpler to learn.

Another innovative experiment is QuantumLink, a new telecommunications network to be operated jointly by Commodore International and Control Video Corporation. The official launch date for QuantumLink was scheduled for October 1, although testing has been going on for several months.

"What we'll be doing, initially for the Commodore 64 and 128, is offering a set of services, mostly on a flat-fee basis for $9.95 a month," says Stephen Case, vice president of marketing for Quantum Computer Services. QuantumLink's offerings will include previews of commercial software that can be downloaded, bulletin boards, a computer information center, news, teleshopping, and interactive telegaming with full-color graphics, says Case. "The $9.95 a month includes communications charges for some of the services—like the encyclopedia, for example. You can use it [Grolier's *American Academic Encyclopedia*] all you want and there's no extra charge."

Some services, such as software downloading and the Chat feature—an interactive online conversation—cost an extra six cents a minute. QuantumLink can be accessed through the Uninet carrier network.

Computer owners who register for QuantumLink before the end of 1985 will get Quantum's special terminal software without charge plus a free month of access. After January 1, the signup fee will be $25, says Case. The special software is necessary because QuantumLink has a graphics interface similar to that of the Macintosh, and telegames such as chess, backgammon, and hangman—which feature full-color graphics and sound—are stored on the disk. (To register online for a free trial, call 1-800-833-9400.)

Online Previews

Commercial programs are not the only products that can now be previewed online. On CompuServe, science fiction fans can read chapters from new books published by Baen Books. There's no charge other than the usual CompuServe connect fees. CompuServe subscribers can reach the Science Fiction and Fantasy Forum by typing GO HOM 29. Baen Books is currently in the forum's Data Library 3 (although that may change by the time you read this). To enter that library, type DL3 and hit RETURN or ENTER. Then type BRO to browse through the various filenames. Subcommands let you retrieve and read a file. You can even comment on what you've read by leaving a message for Baen Books via its CompuServe user number: 70307,541.

The Baen Books files can be read, copied, and distributed freely, as long as they aren't altered or sold. Local bulletin boards can retrieve the files from CompuServe and offer them to their members without charge.

These experiments and others are changing the ways in which people use their personal computers. In two particular areas—immediate acquisition of information and communication among like-minded individuals—telecomputing is becoming easier every day, says Matthew Lesko, an authority on the use of electronic databases and president of Information U.S.A., a database information company.

"Now I can hook up my computer terminal and be on the floor of the stock market even 5,000 miles away. That's a wonderful application."

Immediate communication among members of different professions has already become a commonplace event, reaping extraordinary results, Lesko adds. "It's like-minded people communicating, getting together and talking. It's how our society takes leaps and bounds." ©

Ben's Mom also bought Bowater's 500 sheet package of computer paper with a FREE Christmas Window Ornament that looks like stain glass. She knows Bowater gives her family ***MORE*** than just computer paper.

Ask for Bowater Computer Forms at your local office supply, computer dealer, discount or specialty department store. Or, call us and we'll help you locate the Bowater dealer nearest you.

Bowater *Paper & Files* come in 500, 1000, and 2900 sheet packages in 9½x11 - 20# bond computer paper.

Bowater *Better Image* brand with Christmas Window Ornament comes in 500 and 1000 sheet packages of 9½x11 20# bond computer paper.

BOWATER
BOWATER COMPUTER FORMS INC.
800/527-3412 outside Texas • 214/578-2000
P. O. Box 869020 • Plano, TX 75086-9020

An Overview Of Telecommunications Software

The following chart contains information on a variety of telecommunications programs for several different computer systems. There are hundreds more available, but we have limited this guide to software in the under-$100 price range.

Choose carefully when shopping for a terminal program. The most expensive, multifeatured modem is helpless without adequate software.

Data for this guide was supplied by .MENU—The International Software Database Corporation. For further information and ordering, contact .MENU, 1520 South College Avenue, Fort Collins, CO 80524. Call toll-free 1-800-THE-MENU or 303-482-5000 (in Colorado or outside the U.S.). Telex ISD 454590. When ordering, please use the International Standard Program Number (ISPN).

Product	Price	ISPN	Publisher/ Vendor	Systems	Description
Apple Sourcelink	$29 95	74737-0500	Source Telecomputing Corp	Apple II, II+, IIc, IIe	Communications software designed to supplement use of The Source
ASCII Express II	$59 95	75100-2100	Roger Wagner Publishing	Apple II, II+, IIe	
Basic Terminal	Cassette $14.95	17512-0600	Practicorp International	VIC-20	Allows the user with either a plug-in modem or RS-232/modem combination to communicate with a remote time-sharing system
BITS (Basic Interactive Terminal Software)	$54 95	73612-1000	Software Sorcery	Apple II+, IIc, IIe	
Busiterm	$59 95		Skyles Electric Works	Commodore 64	
CHAT	$40	45537-1000	Lovells	Apple II, II+	
COMMTALK Ver 2	$89.95	29393-1000	Enhanced Technology Assoc.	IBM PC	Has automated communication and information retrieval
Copylink PC	$99 95	84616-1000	U S Digital Corp	IBM PC	
Copylink Ver 2 41	$99 95	83208-2000	U.S. Digital Corp	Apple II, IBM PC	
CW/Term Ver. 1.0	$60	13300-0500	The Code Works	IBM PC	
Data Capture IIe	$90	74850-1100	Southeastern Software	Apple IIc, IIe	
Data Capture Ver. 5 0	$90	74850-1050	Southeastern Software	Apple II, II+	
Data Express	$75	50500-0970	Microlab Inc	Apple II, II+, IIc, IIe	Has an unattended answer mode
Datalink	$99.95	44850-2000	Link Systems	Apple II, II+, IBM PC	
Datalink (enhanced version)	$99 95 $175 00	44850-2500	Link Systems	Apple II, II+, IBM PC	
Direct Connect	$95	25975-1000	Direct.Aid	IBM PC	
Dow Jones Spreadsheet Link	$99	26725-4000	Dow Jones & Company Inc	Apple II, II+, IIe, Macintosh, IBM PC	Download information from Dow Jones News/Retrieval directly into a spreadsheet template set up for analysis
Dow Jones Straight Talk	$95	26725-4250	Dow Jones & Company Inc	Mac	Designed to help the user obtain, store, and organize information from Dow Jones News/Retrieval
Flex-I-Term	$95.95	70675-2000	Source View Corp	Apple II, II+, IIe	
Genterm Ver 2 60	$79.95	37600-1000	Information Analysis Sys. Corp.	IBM PC	Asynchronous communications system with optional terminal evaluation
Habacom	$69 95	33987-0500	Haba Systems Inc.	Mac	
Hello Central	$99 95	67731-2700	Howard W Sams and Company Inc.	Apple II	
Home Connection	$49.95		Penguin Software	Apple II	
HomePak	$49 95	07075-295	Batteries Included	Commodore 64	Integrated telecommunications-database-word processor

TELECOMPUTING
It's only a phone call away.

MacTalk: Telecomputing on the Macintosh
Sheldon Leemon
Arlan Levitan

A complete guide to telecomputing on the Macintosh from choosing a modem and software to accessing information services and electronic bulletin boards.
$14.95 ISBN 0-942386-85-X

COMPUTE!'s Telecomputing on the IBM
Arlan R. Levitan
Sheldon Leemon

The ins and outs of telecomputing on the IBM PC or PCjr, selecting a modem and evaluating terminal software, how to go online with the major information services.
$14.95 ISBN 0-942386-96-5

COMPUTE!'s Telecomputing on the Commodore 64
Edited

Introduces readers to telecommunications, with sections on buying and using modems, accessing information services and bulletin boards, and uploading and downloading files. There is also a disk available which includes the programs in the book.
$12.95 ISBN 0-87455-009-2

Telecomputing lets you call up computers around the world through a network of telephone lines.

To get you started in telecomputing, COMPUTE! Books offers you five top-selling books. Written for the Apple II-series, Commodore 64, IBM PC and PCjr, and Macintosh, the books give you all the information you need, from selecting software to dialing large databases.

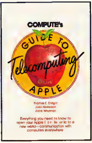
To order your complete guide to telecomputing, give us a call. In the U.S., call toll free 800-334-0868 (North Carolina residents call 919-275-9809).

COMPUTE!'s Personal Telecomputing
Don Stoner

This comprehensive general guide to the world of telecomputing shows how to access databases, receive software, and communicate with others using a personal computer.
$12.95 ISBN 0-942386-47-7

COMPUTE!'s Guide to Telecomputing on the Apple
Thomas E. Enright
Joan Nickerson
Anne Wayman

An informative, easy-to-understand guide to telecomputing on the Apple: covers everything from selecting hardware and software to accessing large databases.
$9.95 ISBN 0-942386-98-1



COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies
324 W. Wendover Avenue Suite 200 Greensboro, NC 27408 919-275-9809
Publishers of COMPUTE! COMPUTE!'s Gazette COMPUTE!'s Gazette Disk COMPUTE! Books and COMPUTE!'s Apple Applications

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England.

SOME HISTORIC BREAKTHROUGHS DON'T TAKE AS MUCH EXPLAINING AS COMPUSERVE.

But then, some historic breakthroughs could only take you from the cave to the tar pits and back again.

CompuServe, on the other hand, makes a considerably more civilized contribution to your life.

It turns that marvel of the 20th century, the personal computer, into something useful.

Unlike most personal computer products you read about, CompuServe is an information service. It isn't software. It isn't hardware. And you don't even have to know a thing about programming to use it. You subscribe to CompuServe—and 24 hours a day, 7 days a week, it puts a universe of information, entertainment and communications right at your fingertips.

A few of the hundreds of things you can do with CompuServe.

COMMUNICATE

EasyPlex™ Electronic Mail lets even beginners compose, edit, send and file messages the first time they get online. It puts friends, relatives and business associates—anywhere in the country—in constant, convenient touch.

CB Simulator features 72 channels for "talking" with thousands of other enthusiastic subscribers throughout the country and Canada. The chatter is frequently hilarious, the "handles" unforgettable, and the friendships hard and fast.

More than 100 Forums welcome your participation in "discussions" on all sorts of topics. There are Forums for computer owners, gourmet cooks, veterinarians, pilots, golfers, musicians, you name it! Also, Electronic Conferencing lets businesses put heads together without anyone having to leave the shop.

Bulletin Boards let you "post" messages where thousands will see them. You can use our National Bulletin Board or the specialized Bulletin Boards found in just about every Forum.

HAVE FUN

Our full range of games includes "You Guessed It!", the first online TV-style game show you play for real prizes; and MegaWars III, offering the ultimate in interactive excitement. And there are board, parlor, sports and educational games to play alone or against other subscribers throughout the country.

Movie Reviews keep that big night at the movies from being a five star mistake.

SHOP

THE ELECTRONIC MALL™ gives you convenient, 24-hour-a-day, 7-day-a-week shopping for name brand goods and services at discount prices from nationally known stores and businesses.

SAVE ON TRIPS

Travelshopper℠ lets you scan flight availabilities (on virtually any airline—worldwide), find airfare bargains and order tickets right on your computer.

Worldwide Exchange sets you up with the perfect yacht, condo, villa, or whatever it takes to make your next vacation *a vacation*.

A to Z Travel/News Service provides the latest travel news plus complete information on over 20,000 hotels worldwide.

Product	Price	ISPN	Publisher/ Vendor	Systems	Description
ICOMM	$100	71550-1000	Smith Educ. Engineering Ser Inc	IBM PC	
Intelink	$59.95	27050-4600	Dynacomp Inc.	Atari 24K RAM	
Kimber-Link	$29.95, plus $3 shipping		Kimbertek, Inc.	Commodore 64	Compatible with auto-dial modems
Kwik-Phone	$19.95	23700-0540	Datamost Inc.	Commodore 64	Has automatic answer, built-in phone book, and bulletin board
MacMail	$59.95	01718-6000	Aegis Development Inc	Mac	Send or receive electronic mail (data/program files) automatically. Interact with Apple-Talk and other networks or person to person
MacTerminal	$99	03900-4150	Apple Computer, Inc.	Mac	
Micro Link II	$99	25400-7600	Digital Marketing	IBM PC	
Micro Link II with Newsnet	$99	25400-3700	Digital Marketing	IBM PC	Will auto-dial, auto-logon, and capture onto disk any information retrieved
Modem-86	$89	18600-1250	Compuview Products Inc.	IBM PC	
Mouse Exchange Terminal	$39.95	26769-5500	Dreams of the Phoenix Inc	Mac	
MTerm Ver. 1.40	$79.95	51537-6000	Micro-Systems Software Inc.	IBM PC	
Omnicomm	$39.95 $34.95	73925-2000	Oakridge Micro Computer	Commodore 64, VIC-20	
Online	$89.95	75100-4400	Roger Wagner Publishing	Apple II, II+, IIe	
OwlTerm	$50	59000-5500	Owl Micro-Communications Ltd.	Apple II	Asynchronous communications package allowing the Apple to behave as a dumb teletype terminal
PC-Dial	$25	09856-1000	Jim Button	IBM PC	
PCModem	$49.95	74412-1000	Solution Software Systems	IBM PC	
PC-Talk Version 2.0	$35	34987-1000	The Headlands Press Inc	IBM PC	
PC-Talk III	$35	31418-1000	Freeware	IBM PC	
PFS Access Ver. A.0	$70 $95	73300-1250	Software Publishing Corp	Apple IIc, IIe, IBM PC	
Pits (PASCAL Interactive Terminal Software)	$54.95	73612-2000	Software Sorcery	Apple II, II+, IIc, IIe	
Pretty Good Terminal	$35	13087-6500	Club Mac User Group	Mac	Has Mac-to-Mac and XMODEM (Christensen) file transfer protocols
Procom-M Ver. 1.6	$99	63273-4000	Prometheus Products Inc.	Apple II, II+, IIc, IIe, Mac	Has XMODEM protocol, phone directory, log-on macro capability, XON/XOFF, capture buffer, text editor, phone log
PTP 1.1	$69.95	82790-1000	Trutec Software	Apple II, IIe	
Reach/86	$39.95	73950-4190	The Software Toolworks	IBM PC	
ReadiTerm	$75	65378-2000	Readiware Systems, Inc.	IBM PC	
Sixth Sense	$89.95	53625-0750	Microtechnic Solutions, Inc	Commodore 64	Communications package with the capability to perform complex tasks including decision-making
Skiwriter II	$49.95	70387-665	Prentice-Hall	Commodore 64	Cartridge-based telecommunications/word processor
Smart 64 Talking Terminal	$39.95	53625-0950	Microtechnic Solutions, Inc	Commodore 64	Operates with the Comvoice Speech Synthesizer for visually impaired people
Smart 64 Terminal +3	$39.95	53625-1000	Microtechnic Solutions Inc.	Commodore 64	
Smart 64 Terminal +4	$39.95	53625-1050	Microtechnic Solutions Inc	Commodore 64	
SourceLink	$49.95	74737-1000	Source Telecomputing Corp.	IBM PC	Communications package designed by The Source for communication to this and other time-sharing services
Syncomm	$44.95	77500-7880	Synapse Software	Atari 400/800, XL, XE	
Talking Termexec	$95	30345-1000	Exec Software, Inc.	Apple II, II+, IIc, IIe	A communications package for the visually impaired
Telelink I	16K-$29.95 8K-$24.95	05750-8400	Atari Corp.	Atari 400/800, XL, XE	
Telelink II	$19.95	05750-8500	Atari Corp.	Atari 400/800, XL, XE	

Product	Price	ISPN	Publisher/ Vendor	Systems	Description
Tele-Porter	$79.95	69200-6000	Sensible Software, Inc.	Apple II+, IIe	
TeleText	$49 95	53425-9000	Microsparc, Inc	Apple II, II+, IIe	
Teletext Ver 1.0	$79 95	51500-6000	Microsparc, Inc	Apple II, II+, IIe	
Telpac Ver 2-0	$99	84619-1000	U.S. Robotics, Inc.	Apple II, IIe, III	Can automatic call, logon, transfer files, and make timed calls
Telstar 64	$29 95		Eastern House Software	Commodore 64	
TermExec	$95	30345-2000	Exec Software, Inc	Apple II, II+, IIc, IIe	
TermExec	$79 95	64475-4000	Quinsept, Inc.	Apple II+, IIc, IIe	
Terminal II	$60	80950-6900	Telephone Software Connection	Apple II+, IIc, IIe	Features auto-logon/logon-memorization, automatically records online sessions, review, print, and save to disk
Transend	$89	75500-1000	Transend Corporation	Apple II+, IIe, IBM PC	
Videolink 88	$59 95	86878-1000	Windmill Software, Inc	IBM PC	
VIP Terminal	$59 95		Softlaw, Inc.	Commodore 64	
XL	$75	03184-9000	AML	Commodore 64	
Z-Term	$99 95	75100-8000	Roger Wagner Publishing	Apple II, II+, IIe	

What's New Online?

Kathy Yakal, Assistant Features Editor

The major telecommunications services have added several new features over the past year, and a few new services oriented toward personal computer users have come online. Here are the highlights.

American People/Link

Last December, American Home Network premiered American People/Link, a telecommunications network focusing on family entertainment and online conversations. Electronic mail, a CB simulation, and a wide variety of telegames are its main features.

In mid-August, American People/Link started adding online clubs to its other services. Similar to special interest groups (SIGs) on other telecommunications networks, they provide an electronic forum for people with similar interests to share information. Initial clubs cater to such interests as sports, aviation, humor, women's issues, and health.

Subscriber fees are as follows: For the first three hours of nonprime-time use each month, the hourly charge is $4.78 for 300 bps and $7.78 for 1200 bps. Additional time costs $2.95/hour for 300 bps and $5.95/hour for 1200 bps. Prime-time access is $9.95 for both 300 and 1200 bps ($14.95 in some cities).

For more information, contact: American Home Network, Inc., Arlington Ridge Office Center, 3215 N. Frontage Road, Suite 1505, Arlington Heights, IL 60004. 800-524-0100 (Illinois residents call 312-870-5200).

CompuServe Information Service

CompuServe, the nation's largest consumer information service, experienced tremendous growth in 1985. Its subscriber base grew by more than 70 percent to nearly a quarter-million, and several new services were added.

Travelshopper gives subscribers access to Trans World Airlines' reservation system. You can find the lowest rates and most convenient flights, then make a reservation while online. Tickets can be sent to your home or to the airline ticket counter, or issued by a local travel agency.

The *Executive Service Option* (formerly called *Executive Information Services*) is a database of sophisticated financial information

which is now available to all subscribers. It offers a variety of tools for investment and financial planning, as well as special merchandise offers and discounts. There is a one-time charge of $10 ($5 for new users) and a $10 minimum monthly usage fee.

CompuServe has also upgraded and simplified its electronic mail service. *Easyplex* features different modes for different levels of expertise; online instructions; easy editing; and an "address book" which stores names and user IDs of up to 50 people.

Gannett Co., Inc., is now distributing *USA TODAY Update* through CompuServe. *Hotlines*, updated hourly from 8 a.m. to 11 p.m., offers business, financial, local, and international news, plus weather reports. *Decisionlines*, updated daily, is targeted to specific industries and professions such as travel, technology, law, and energy.

Since August 1983, the NCR Universal Credit Union has allowed its members to conduct transactions electronically from anywhere in the world through CompuServe's *Companion at-Home*. In the last year, three additional major credit unions have announced an intention to do the same: Northwest Orient Airlines Employee Credit Union, Pacific IBM Employees Credit Union, and Oak Ridge National Laboratories Employees Credit Union.

For more information, contact: CompuServe, P.O. Box 20212, Columbus, OH 43220. 800-848-8199.

Delphi

Since June, Delphi has offered service at 2400 bps, for an additional $5 an hour.

Two new areas of the service have also been developed. Subscribers can now get current news, sports, and financial information on Delphi through AP News Services. And owners of Commodore, Apple II-series, Macintosh, and Atari computers can share information and get technical help through several new online SIGs.

For more information, contact: Delphi, 3 Blackstone Ct., Cambridge, MA 02139. 800-544-4005.

Dow Jones News/Retrieval

Dow Jones has added two new databases to its information service. *Peterson's College Selection Service* has profiles on more than 3,000 two- and four-year colleges and universities. And a new medical and drug reference database addresses the diagnosis of hundreds of diseases and offers information on many pharmaceutical drugs.

American Express Advance lets cardholders look up previous statements on their accounts. *American Express Shopping And Travel Service* offers online shopping and travel information.

In June, Dow Jones' per-minute fees for 300 bps changed to 90 cents (prime time) and 20 cents (nonprime time). The 1200 and 2400 bps rates are double the 300 bps rate. Certain business-related databases require an additional 30 cents (prime) and 60 cents (nonprime) per minute. In addition to the $75 standard membership fee, there's also a $12 annual service fee.

For more information, contact: Dow Jones News/Retrieval, P.O. Box 300, Princeton, NJ 08540. 800-257-5114.

The Source

Over the past year, The Source simplified use of its telecommunications network. The updated menu incorporates a self-teaching design to help users find what they're looking for more quickly.

Online assistance has always been available on The Source, but now it's expanded and it's free. The tutorial includes four lessons of graduated difficulty to familiarize new users with the system. Unlimited free access to this assistance allows both new and experienced subscribers to explore areas of the system that they may not have known about before.

In August, officials at The Source announced that individual SIGs would soon be online. Though details have not been fully developed at this writing, the SIGs are expected to address the special interests of personal computer owners. An additional per-minute fee will be charged for this service.

In August, 2400 bps service began in ten major cities. Additional cities will soon be added via Uninet and Telenet. The base rate for prime-time 2400 bps service is 46 cents per minute; nonprime time is 20 cents per minute.

A new database contains updated listings for 14,000 domestic and 8,000 international hotels. Each listing contains the hotel's address and telephone number, as well as information on restaurants, convention facilities, sports and leisure services, and rates.

For more information, contact: The Source, 1616 Anderson Road, McLean, VA 22102. 800-336-3366.

Viewtron

Viewtron is a new videotex service scheduled to begin this fall for Commodore, Apple, and IBM owners. Operated by Viewdata Corporation, a subsidiary of Knight-Ridder Newspapers, Inc., Viewtron was to start October 1 in most areas of the U.S. with access to a Telenet, Tymnet, or Uninet number, except Massachusetts, New Hampshire, Vermont, and Maine. Viewtron plans to offer news, weather, sports, and current stock prices; book, movie, and software reviews; communication with other subscribers through electronic mail and a CB simulator; and online shopping and banking.

Viewtron is to be the first major news and information service in the U.S. to display color graphics, though only for Commodore users. Because of this feature, Commodore owners need special terminal software designed for the system. IBM and Apple owners can use any terminal software with VT-100 emulation (or Viewtron's package).

To subscribe, you must buy a Viewtron Software Starter Kit ($9.95) which contains terminal software, one free hour of service, an ID and password, and a user manual. Rates after the first hour are nine cents a minute (after 6 p.m. weekdays, all day weekends) and 22 cents a minute (weekdays before 6). There is no monthly minimum and no extra charge for 1200 bps access.

Viewdata is offering free starter kits with the purchase of some Anchor Automation modems. A 300 bps Westridge 6420 modem with software is $49.95; a 1200 bps Volksmodem 12 is $189.95.

For more information, contact: Viewdata Corporation of America, Inc., 1111 Lincoln Road, 7th Floor, Miami Beach, FL 33139. 800-543-5500, Department 9401. ©

HOW TO GET OVER $2000 WORTH OF NEW CAPABILITIES FOR YOUR COMMODORE 64™

FOR $599*

The Spartan™ is the Apple™ II+ emulator for your Commodore 64™ that will open up a whole new world of hardware and software for you! Imagine adding these features to your Commodore 64™ for the Spartan™ price of $599: ☐ Apple™ II+ hardware and software capabilities ☐ 64K RAM expansion ☐ four software selectable Commodore 64™ cartridge slots ☐ non-dedicated 8-bit parallel port ☐ standard audio cassette deck capabilities for your C-64™.

The suggested retail value of comparable products offering only these capabilities is over $2200.00* — but the Spartan™ gives you much, much more! By building on your investment in your Commodore 64™— an excellent introductory computer — you create a whole new system with both C-64™ and Apple™ II+ capabilities. There is a whole other world out there! The huge selection of Apple™ II+ hardware and software is now yours to explore! Call toll free for the Spartan™ dealer nearest you.

*All prices quoted are in U.S. funds, freight and taxes not included. Value of components equivalent to the Spartan™ system are quoted from Apple™ II+ CPU and Apple™ II+ single disk drive 1983 list prices, and from current suggested list prices and component specifications of other peripheral manufacturers. Commodore 64™ and Commodore logo are trademarks of Commodore Electronics Ltd. and/or Commodore Business Machines, Inc. Apple™ II+ is a trademark of Apple Computer Inc. Spartan™ is a trademark of Mimic Systems Inc. and has no association with Commodore Electronics or Apple Computer Inc. The Spartan™ is manufactured by Mimic Systems Inc. under license granted by ATG Electronics Inc. of Victoria, B.C. Canada

MIMIC

FOR INFORMATION WRITE:
MIMIC SYSTEMS INC.
1112 FORT ST., FL. 6N
VICTORIA, B.C.
CANADA V8V 4V2

TO ORDER CALL:
1-800-MODULAR
(663-8527)

Programming Languages

Communicating With Your Computer

Todd Heimarck, Assistant Editor
Kathy Yakal, Assistant Features Editor

BASIC, Logo, Pascal, FORTRAN, COBOL, Forth—these and a hundred other odd-sounding names are the languages we use to communicate with computers. Since the beginning of the computer age, scientists and programmers have been striving to make this human-machine interaction easier, faster, and more powerful. Why are there so many languages? Which are best? And what will tomorrow's languages be like? The answers may help you decide which language is best for you.

Imagine that if every time you asked someone to do something, you had to tediously explain each step of the procedure. Outlining something as simple as taking out the garbage could take ten minutes:

Walk to the garbage can by placing one foot in front of the other and moving forward. Stop. Bend over at the waist, extending both arms out in front of you. Put your hands on either side of the garbage bag and grasp it by curling your fingers and applying pressure to get a good grip. Stand up straight again, holding the bag in front of you. Turn around and face the opposite direction, still holding the bag. Walk toward the back door. Stop. Bend over again and release your grip on the bag. Stand up, extend your right arm, and grasp the doorknob. Apply pressure, twist, and pull until the door opens.... **And so on.**

Human beings don't need that kind of step-by-step instruction for most tasks. But computers require it for *all* tasks. Technically, the only way to make a computer do something is to rearrange its internal pathways of electricity by flipping the equivalent of thousands of microscopic on/off switches. By programming at the computer's most fundamental level, —a binary code of ones and zeros which controls those switches—programmers can instruct computers to carry out very simple tasks, like adding two numbers or storing a number in memory. When hundreds or even thousands of these simple commands are combined to form a program, computers can seem to handle tasks of great complexity.

But programming a computer in binary codes can be a daunting job. To make it easier and faster, computer scientists and engineers have spent the last four decades developing scores of programming languages as alternatives to communicating with computers on the binary level. Many of these languages are composed of familiar English words, and they serve as translators or interpreters between the language of the programmer and the language of the machine. For example, many of today's personal computers come with a language called BASIC, which stands for *Beginner's All-purpose Symbolic Instruction Code.* A typical English-like BASIC command is PRINT. When PRINT is followed by some text inside quotation marks, such as PRINT "HELLO", the computer prints the text on the monitor screen. To do the same thing directly in machine language, a programmer might have to write a half-dozen or more commands.

For this reason, languages such as BASIC are known as *high-level languages*—they are relatively far removed from the binary level of the machine. Programming in a high-level language versus programming in machine language is somewhat like the difference between saying "Please take out the garbage" or outlining the whole process step-by-step as shown above.

There are other reasons why high-level languages are continually being developed, too. Different people have different programming styles, so more languages provide more choices. Also, special programming jobs require specialized tools; the language for writing an accounting program might not be the best for writing an adventure game.

Special programming jobs require specialized tools; the language for writing an accounting program might not be the best for writing an adventure game.

The evolution of these languages, however, has distanced programmers from the inner workings of computers. High-level languages make it easier to write programs, but fewer and fewer people understand what's really happening inside the box—how the electrons are zipping in and out of logic gates. It's like driving a car without thinking about how the gas and air are exploding inside the cylinders, pushing the pistons up and down. Whether or not it's important to know these details is a matter of debate within the computing community.

Today, you can run a program on just about any personal computer without knowing anything about programming. Usually it's as simple as inserting a floppy disk or program cartridge, switching on the system, and perhaps typing a single command to get things started.

This is quite a jump from 40 years ago, when the first electronic digital computer, ENIAC, was built. ENIAC (Electronic Numeric Integrator and Calculator) was a 30-ton, 100-foot-long machine which contained almost a hundred thousand vacuum tubes, resistors, and capacitors. ENIAC had to be programmed by *hard-wiring*—engineers rewired it for each new program they wanted to run. There was no memory inside the computer to store programs. And today's mass-storage devices, such as floppy disks and tapes, were not yet imagined. Hard-wiring ENIAC could take days as engineers prepared the monster to solve one type of complex calculation. Once programmed, ENIAC could solve the equations far faster than people. But if a different type of calculation was required, the hard-wiring had to start all over again.

The difficulty of programming a behemoth such as ENIAC meant that only a handful of scientists and engineers could really "talk" to the computer. And they had to communicate completely in the machine's own primitive language of wires and connections.

In addition to being enormously expensive to build and maintain, these early computers were expensive to use because hard-wiring took so much time—time that could be spent on calculations. So engineers borrowed an idea from computer pioneer John von Neumann—*stored programs*. Adding memory to a computer to temporarily store a program as it runs is much faster and easier than rewiring the hardware. You can change programs simply by replacing the program in memory with a new one.

By mid-1948, British computer scientists had completed the Mark I, commonly recognized as the first stored-program computer. By flipping switches on the front of the Mark I, engineers could enter short programs into the machine. This was a major improvement, but still clumsy. Reportedly, the codes had to be entered *backward.*

Next, a way had to be found to store programs between jobs; there isn't nearly enough memory in a computer to permanently keep all the possible programs that could be written. Also, many programs require data

© 1985 LFL

You've just discovered the Eidolon—a curious 19th century machine whose inventor vanished without a trace. Only his journals and sketches remain. They tell of an incredible magical realm—a maze of caverns populated by strange creatures noted as Greps, Biter Birds and Bottlenecks. And "intelligent" Guardian Dragons—who hurl colored fireballs of energy?

The machine itself belies its quaint Victorian charm. For the Eidolon glows with the power of enchanted energy. It awaits, pristine and gleaming—perfectly preserved for over a hundred years by the powerful forces that propelled it and its pilot to another dimension.

With the fascinating first-person point of view, you can climb into the pilot's chair and fly this mysterious magical machine. And the haunting fractal graphics take you deeper and deeper into an endless maze of mystical caverns.

An adventure so real, it'll make you wonder: What ever happened to the Eidolon's mysterious inventor? Only the adventurous of spirit will know his fate. The Eidolon—scientific curiosity, or passport to a magical dimension?

	C64/128	ATARI	APPLE
The Eidolon	✓	✓	✓

EPYX® COMPUTER SOFTWARE
1043 Kiel Ct., Sunnyvale, CA 94089

Strategy Games for the Action-Game Player®

© 1985 LFL.

© 1985 LFL

* See specially marked boxes for details No purchase necessary Sweepstakes ends Dec 31 1985 Official rules available at participating dealers

which changes from job to job and can't be stored as part of the program, such as the monthly electric bills of utility company customers.

This time, engineers borrowed a piece of nineteenth-century technology—cardboard punch cards. This idea was originally developed by nineteenth-century mathematician Charles Babbage, who took the concept from an earlier system used by the French to control weaving looms. Punch cards had proven their worth in data processing during the 1890 U.S. census, when they were used to speed up tabulation on mechanical adding machines.

By adapting punch cards to computers, it became possible to write and store programs without tying up the machine itself. Programmers typed their programs on keypunch machines, then waited their turn to feed the stack of cards into the computer. After the results were printed out, the computer was prepared to accept another batch of cards. This system was called *batch processing*.

For the first time, programmers were physically separated from computers. There were software experts, who wrote programs on batches of cards, and hardware experts, who fed the cards into computers.

The first real software breakthrough was an *assembler* program. An assembler translates mnemonics like LD (load a number from memory) and ST (store a number in memory) into the binary ones and zeros the computer understands. Each assembler operation code (or *opcode*) corresponds directly to a machine language instruction.

Soon, programmers began collecting useful pieces of programs written with assemblers. For example, if someone needed a routine to calculate square roots, they could borrow one from another programmer who had already figured out the math, rather than waste time reinventing the wheel. Such a fill-in-the-blanks routine is called a *macro-instruction*, or macro for short.

A library of macros isn't quite a language, because it's not organized or standardized. But macros were the first step toward high-level languages.

As computer education began seeping downward from colleges, for the first time there was a need for languages tailored especially for young people.

One of the first high-level languages was FORTRAN (FORmula TRANslator), developed in 1954. Before FORTRAN, engineers and scientists who were unfamiliar with computers had to describe a problem to a computer programmer, who would then write a program to solve it. FORTRAN made it easier for scientists and engineers to write their own programs.

Just as FORTRAN was written for engineers, COBOL (*COmmon Business Oriented Language*) was created for accountants. Developed in the 1950s by U.S. Navy Captain Grace Hopper, COBOL is still one of the most popular languages for large business computers, and is often used to write payroll programs and other applications in large data processing departments.

In 1964, when FORTRAN and COBOL were the most popular programming languages, two Dartmouth University professors formulated a couple of important ideas. First, they suggested that instead of processing programs in batches, a single computer could be hooked up to several terminals, sharing its time among many users. A fast typist works at perhaps 100 words per minute, while a computer can accept keystrokes much faster—in millionths of a second. A *time-sharing system* of terminals would allow more than one person to use the computer simultaneously. Because the computer works so fast, each person could have the illusion that he was the only one working with the machine.

Their second idea was a new language, BASIC, a general computing language which would be easier to learn than FORTRAN or COBOL and more flexible.

Dartmouth became the first university to make computer time generally available to undergraduates, thanks to time-sharing and BASIC. (The two professors, John Kemeny and Thomas Kurtz, recently released a new version of BASIC called *True BASIC*.)

With batch processing, programmers had to write a program by punching it onto cards, then submit it for processing, collect the results the next day, find out there was a bug, rewrite it, submit it again, and so on. Time-sharing allowed programmers to begin debugging a program immediately. It also made computers accessible to more people and paved the way for personal computing.

Soon after BASIC was developed, many more programming languages began appearing. Computers were being adapted to more applications, and more people began using computers, so demand grew for better and more specialized languages.

In the late 1960s, a debate heated up within the academic and computer communities over *structured programming*. This is a method intended to keep programmers more organized and programs more readable and easily modified. The first language specifically designed to encourage structured programming was Pascal—invented by Niklaus Wirth in Switzerland and named after the French mathematician and logician, Blaise Pascal. Today, Pascal is popular in high schools and colleges because instructors say it teaches good programming style. It's also easier to follow the flow of a program written in Pascal.

Meanwhile, computer education began seeping downward from colleges into high schools, junior highs, and even elementary schools. For the first time, there was a need for languages tailored especially for young people. In the late 1960s, Seymour Papert of the Massachusetts Institute of Technology

© 1985 LFL

It takes all kinds to make a galaxy interesting. Like you, a *techno-scavenger.* Your kind are out to get rich scavenging Ancient technology. And now you've found the fabled Koronis Rift—the weapons testing grounds of the Ancients. The chance of a lifetime awaits you.

Abandoned war hulks litter the Rift—crammed with exotic weapons and technology. The lifelike fractal graphics take you to this mythical land of the Ancients. The mind's-eye point of view puts you right in the driver's seat of a Surface Rover. A hulk looms before you—your mind races, feverishly planning a strategy. What weapons do you need to survive? What technology will fetch the highest price? If you can pack your battered Rover full—you'll be rich beyond your wildest dreams.

But it won't be easy. The Guardians—genetically engineered mutants—stand watch over their creators' technology. And they deal swiftly and ruthlessly with characters like techno-scavengers.

But you've got a plan. If you and your trusty Science Droid can scavenge the right combination of weapons and technology—and get off the planet alive—you'll make it big. Destroy the Guardian base and you'll even be a hero! The treasures of the Ancients are yours . . . if you've got what it takes.

	C64/128	ATARI	APPLE
Koronis Rift	✓	✓	✓

EPYX® COMPUTER SOFTWARE
1043 Kiel Ct., Sunnyvale, CA 94089
Strategy Games for the Action-Game Player®

© 1985 LFL

© 1985 LFL

* See specially marked boxes for details No purchase necessary Sweepstakes ends Dec 31 1985 Official rules available at participating dealers.

developed Logo. Many of Logo's commands give directions to a *turtle* on the computer screen, a small object whose movements define and execute a graphics program. The onscreen turtle was adapted from Papert's original Logo, which attached the computer to an actual robotic turtle which children could program to draw designs on paper. Many elementary schools now teach Logo as the first programming language for young children.

New approaches to programming languages also were being explored. For example, Forth is an unusual language originally developed to control telescopes in observatories. It's roughly halfway between machine language and high-level languages like BASIC, and is *extensible*—you can define new functions and commands which then become part of the language. In a sense, it's a language that lets you create your own personal language. If you want, you can build up the language piece by piece, until you finally define a single word that runs the whole program.

Although there are hundreds of programming languages, most are not available for personal computers. Some languages were designed for large mainframe computers and cannot fit into small amounts of memory. Others are just too specialized for general use. If you'd like to explore the alternatives, here are some issues to consider:

• *What types of programs will you be writing?*

One language might offer lots of commands for handling files and variables, but very little in the way of graphics. Another might be strong in mathematical functions, but weak in handling strings and text. Look for a language that is suited for the kind of programs you want to write. There are always books and manuals which list the commands available in a language and describe what they do.

• *How much control do you want over the hardware and software? Is the language high-level, low-level, or somewhere in between?*

A low-level language like machine language puts you in direct control of the computer. Individual instructions do very simple things,

The commands in high-level languages look more like words in a human language, so they're easier to learn.

like fetching and storing numbers in memory, comparing two numbers, and basic addition and subtraction. To multiply two numbers, you might need several instructions. (However, on the newer chips, multiplication requires only a single instruction.)

High-level languages take you several steps away from machine language and the hardware. The commands look more like words in a human language, so they're easier to learn. Also, individual commands are usually broader, performing tasks which might require dozens of commands in machine language. But you pay a price: Direct control over the finer points of the computer may be more difficult, and the finished programs run more slowly and often consume more memory. Remember, the only language the computer *really* understands is machine language—at some stage, it has to translate programs written in another language into its native tongue.

• *How fast is the language?*

Speed is important in some programs. A certain part of a program may take 1/20 second to execute in one language and 1/2 second in another, not a noticeable difference if it's used once or twice. But if it's executed several thousand times, the difference could become significant.

Machine language is the fastest, and most commercial software is written in machine language. (In fact, most high-level languages themselves are written in machine language.) Mid-level languages such as Forth and C, while not as fast as machine language, are generally quicker than higher-level languages.

Because the faster languages are usually low-level, they may be more difficult to learn and use. High-level languages are fine for many programs, and here's where you must strike a balance: Would you rather spend five hours working with a low-level language to write a program that runs in one minute, or spend one hour working with a high-level language to write the same program that runs in 15 minutes? If you're going to run the program every day, you might choose to spend the extra time writing it with the faster language. But for an infrequently used program, you might prefer the language that's easier and slower.

In some cases, the speed of a language doesn't matter. If a printer seems to take forever to print reports or mailing labels, rewriting the program with a faster language may not help. The printer is probably the limiting factor on speed, not the language.

• *What are the system requirements? And how much free memory for programs remains after the language is loaded into the computer?*

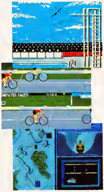
You may find languages that require a certain operating system. C, for example, was originally written for the Unix operating system, although that has changed—other versions of C are now available. On a Commodore 64, certain languages work only with the CP/M cartridge. And some languages won't work without two disk drives.

Check the memory requirements. You may have to install additional memory boards or controller cards. Even if you have the minimum memory specified for a certain language, you may be left with very little space for your programs.

• *What programming style are you most comfortable with? Scientific and structured? Or creative and artistic?*

Some people write programs methodically, step by step. They draw a flowchart on paper, diagramming the program in modules. They fully document each section, describing exactly what happens when. Not until they finish the preliminary planning and structuring do they enter the program into the computer. In business, the structured approach is preferable. If a programmer quits for

22 FAST-PACED, EXCITING EVENTS. GO FOR THE GOLD!



SUMMER GAMES.®
Want some great play action? This award winning best seller brings you eight great events, including Swimming, Diving, Track, Gymnastics, and more. Compete against world records. Or get together with a group of up to eight for some good competitive *fun*. Realistic, graphics and *action* will challenge you again and again to *go for the gold*.

SUMMER GAMES II.™
You asked us for more great events. Here they are. Rowing, Triple Jump, Javelin, High Jump, Fencing, Cycling, Kayaking, even Equestrian. Like SUMMER GAMES, you get spine tingling *action and challenge* for one to eight players. These aren't just games. They're *the games*. And you're the star.

WINTER GAMES.™
You've got to see the graphics, and *feel* the knot in your stomach as your bobsled careens along the walls of the run to know why people are wild over WINTER GAMES. Seven events give you a variety of action—from the endurance of the Biathlon to the timing of the Ski Jump, and more.

What are you waiting for?
Play your favorite events over and over. Play all 22. Set up teams. Challenge your friends. These great action-strategy games are sure to be the stars of *your* collection.

At your local Epyx dealer.

	APPLE	MAC	ATARI	C64/128
Summer Games	✓		✓	✓
Summer Games II	✓			✓
Winter Games	✓	✓		✓

* See specially marked boxes for details. No purchase necessary. Sweepstakes ends Dec. 31, 1985. Official rules available at participating dealers.

EPYX® COMPUTER SOFTWARE
1043 Kiel Ct., Sunnyvale, CA 94089

Strategy Games for the Action-Game Player®

SPINNAKER® THE LEADER IN FAMILY LEARNING SOFTWARE.

From pre-school counting to high-school compositions, Spinnaker offers the high quality educational programs you have come to expect. For example, there's the HOMEWORK HELPER™ series which provides step-by-step methods for improving writing and math skills and helps kids tackle otherwise troublesome school assignments.

TELARIUM.™ INTERACTIVE FICTION AT ITS BEST.

Here's graphic interactive fiction by famous writers of science fiction, fantasy, and mystery like Ray Bradbury, Arthur C. Clarke and Erle Stanley Gardner. Each story unfolds with you in the center of the action–and completely in control. In PERRY MASON: THE CASE OF THE MANDARIN MURDER,™ you are the world famous criminal lawyer. Challenge the evidence. Cross-examine the witness. Only you can prove your client's innocence. And time is running out.

A SOFTWARE OFFER FOR ALL AGES.

Any software offer including something for everyone in the family is certainly fantastic. But combine that with a free product offer and what you have is something spectacular. Look for the specially marked boxes of The Family Software Spectacular at your retailer.

The Family Software Spectacular™

TURN PAGE FOR MORE INFORMATION ON THE FAMILY SOFTWARE SPECTACULAR. →

some reason, the company needs to know how the programs are put together.

Others prefer a looser, more improvisational style. They type a few lines, run the program, make some changes, test it again, and so on. Then they write and test a new section. The programs are not necessarily unstructured or illogical; it's just that the program ideas are not written down. The program may change as it develops, evolving into something quite different from the original plan.

There are advantages and disadvantages to each style. Planning ahead takes a lot of time up front, before a single line is entered. And it locks you into a certain structure. But the programs are generally easy to follow and debug. When teams of programmers work together, they generally use the planned-out approach.

More casual programmers who work interactively with the computer can see immediate results, positive feedback that the program is progressing. There's also an element of creative experimentation: "I'll try this and if it doesn't work, I'll try something else." Less time is spent on planning, and more time on actual programming. The casual approach can be carried too far, however. If the program is written sloppily, even its author might not understand how it works if modifications are required a few months later.

• If you need to write fast programs, but don't want to use machine language, will a compiler do the job?

There are two general ways in which commands in higher-level languages are translated into the machine language that the computer can understand.

An *interpreter* language translates the commands as the program runs, on the fly. The BASICs built into personal computers are interpreters.

A *compiler*, on the other hand, translates all the high-level commands into machine language before running the program. This compilation step may take several minutes, but when it's done, the finished program usually runs much faster than an interpreted program (though not as fast as programs written directly in machine language).

Some languages (including BASIC) are available as both interpreters and compilers. There are tradeoffs either way. Compiled programs run faster than interpreted programs, but usually require much more memory—sometimes too much for small computers. Interpreters are more interactive, because you can type in a few statements, quickly try them out, and continue. A compiler might take ten minutes to compile a program. The choice between an interpreter and a compiler depends a great deal on your personal programming style, the amount of memory in your computer, and your need for speed in the finished product.

Ultimately, the language you choose for communicating with your computer depends on a great number of things. After reviewing all the options, you may find it desirable to learn more than one language, especially if you plan to write different kinds of programs.

During recent years, computer scientists, programmers, and linguists have been working in the field of artificial intelligence to develop methods for computers to more closely mimic human thought. An important part of this work has been research into so-called *natural languages*—those languages which humans use. We may see a day when the perfect natural language interface is developed, and we need only tell the computer, in our own tongue, what we want it to do. The latest generation of personal computers—such as the Apple Macintosh, Atari 520ST, and Commodore Amiga—represent another small step in that direction.

For now, however, control over a computer means meeting the machine at least halfway—learning a language which gives the computer something intelligible to work with. No longer must people learn to program to use a computer enjoyably and productively. But for thousands of computer owners, learning to communicate with their machines in a common language opens up the world of computing in ways which are better experienced than explained.


BATTERIES INCLUDED
"The Energized Software Company!"
We started with Commodore, designing programs that quickly became industry success stories. Now we're moving on, applying our expertise to other systems.
Look for Apple, Atari, IBM, and Commodore software with the Batteries Included label!
Look forward to excellence, in every respect. Power and performance, ease of use, incredibly low prices, for programs that help you in so many ways.
PaperClip
The Consultant
"Performance: excellent... Error-Handling: excellent... Value: excellent"
FAMILY COMPUTING
(a "Billboard" magazine #1 best seller)
"The best Atari word-processor ever."
ANTIC MAGAZINE
"...capable of very large and complicated searches ...a very good system."
TPUG MAGAZINE
CAL-KIT
Keys To TYPING
HomePak
B/GRAPH
"quite simply the best ...the highest rating possible."
ANALOG COMPUTING
B.I.
BI-80
Also from BATTERIES INCLUDED:
Calkit: problem-solving spreadsheet program with built-in templates for the most-needed home and business applications – including income tax, budgets and many more.
B/Graph: professional-quality graphics/charting and statistical analysis package turns your data into superb visuals.
Sales Force: professional sales management and reporting program saves you valuable hours.
"Keys To Typing": easy to follow tutorial program typing teacher.
SpellPack
CHECKBOOK
"Literature Challenge – Introduction To Shakespeare": learn about the bard's three most-taught plays – and have fun! (Other authors on the way.)
B.I.-80 Column Display: add-on module doubles your screen capacity and improves visibility.
HOME ORGANIZER SERIES home database managers. (8 programs including Home Inventory, Recipes, CheckBook, Address Book, Audio/Video Catalogue and more)
BATTERIES INCLUDED
"The Energized Software Company!"
30 Mural Street
Richmond Hill, Ontario
L4B 1B5 CANADA
(416) 881-9941
Telex: 06-21-8290
17875 Sky Park North, Suite P
Irving, California
USA 92714
(416) 881-9816
Telex: 509-139
WRITE TO US FOR FULL COLOUR CATALOGUE of our products for COMMODORE, ATARI, APPLE and IBM SYSTEMS.
FOR TECHNICAL SUPPORT OR PRODUCT INFORMATION PLEASE PHONE (416) 881-9816
SOME PROGRAMS ARE NOT AVAILABLE FOR ALL SYSTEMS
Commodore, Apple, Atari and IBM PC are registered trademarks of Apple Computers, Inc., Atari, Inc., Commodore Business Machines, Inc., and International Business Machines, respectively

commodore

PERSONAL COMPUTERS

NEW! **Commodore 128 Personal Computer System** .... **Call For Price**

1101 Letter Quality Printer	274.95
1702 Monitor	174.95
Commodore 64 Computer	144.95
1541 Disc Drive	169.95

PRINTERS

Same As Silver Reed 400

Cardco LQ/3 Daisy Wheel Printer Letter Quality Printer with Built-in Interface Free Cardco Mail Now, Spell Now, Write Now Included	199.95
Gemini SG 10	209.95
Gemini SG 15	359.95
Gemini SG 10C with Built-in Commodore Interface	Call
New! Gemini SR 10115	Call
New! Gemini SB 10	Call
Silver Reed 400	249.95
Panasonic 1091	Call

MONITORS

Sharp 13" Color Composite Monitor 13M31U	144.95
BMC Color	149.95
Pan 1300 RGB & Comp	289.95
Sharp Green	64.95
Sharp Amber	69.95

DISC DRIVES

CCA Exclusive - New from Commodore!

Enhancer 2000 Disc Drive Faster than Commodore 1541, 100% Commodore Compatible, 2 Yr. Extended Warranty	164.95
MSD Super Disc II	399.95
Single Drive	179.95
Indus GT Disc Drive	249.95

TOMY ROBOTS

Unleash the Power of Your Computer Directly Into the Tomy Robots

Includes •Computer, Interface, and Disk •Create, Edit, Save, Retrieve and Mix Your Programs •For Tomy Omnibot, Omnibot 2000 and Verbot All on the Same Disk — **Call for Price**

Interfaces

Telesys Turboprint GT Prints Enhanced Graphics	59.95
Cardco G+	Call
Microworld	Call
PPI	44.95

Discs

20 Memorex SS/DD w/Storage Case	24.95*

*After Rebate

Diskettes

Nashua	8.95
Scotch	11.95
Maxell	11.95
Memorex	11.95
BASF	8.95

Modems

Mighty Mo	64.95
Commodore 1660	79.95
Total Telecommunications	Call

Wico Joysticks

The Boss	12.88
The Bat	14.88
3-Way	19.88

HOLIDAY SOFTWARE SPECIALS AT UNREAL PRICES!

Software Special # 1

commodore

$8.88

Simons Basic or Commodore's Programmer's Reference Guide

Software Special # 2

ACTIVISION HOME COMPUTER SOFTWARE

$9.99

Decathlon, Pitfall I or II, Zone Ranger, Space Shuttle, Master of the Lamp, Star League Baseball, On Field Football, On Court Tennis, H.E.R.D.

Software Special # 3

ACTIVISION HOME COMPUTER SOFTWARE

$14.88

Ghostbusters, *New!* Fast Trax, *New!* Pet Person, *New!* Hacker, *New!* Alcazar, *New!* Countdown to Shutdown

Software Special # 4

EPYX

$19.99

Fastload, **Wintergames**, **Hot Wheels**, Breakdance, Jet Combat Simulator, G.I. Joe, Summer Games I or II, Barbie, Hot Wheels, Baseball

Software Special # 5

EPYX

JUMP MAN **2.88**

Penguin Software

Quest	19.99
Transylvania	19.99
Sword of Kadash	19.99
Xyphus	19.99
Graphics Magician Picture Painter	21.99

Professional Software

Fleet System 2 - Word Processor with built-in 70,000 Word Spell Check	59.95

CBS Software

Dr. Seuss Fix up The Mix-up Puzzler	19.99
Sesame Street Letter-Go-Round	19.99
Big Bird's Fun House	19.99
The Sea Voyager	24.99
Weather Tamers	12.99
Movie Musical Madness	9.99

Success w/Decimals (Add/Subt) D-T	14.99
Success w/Decimals (Mult/Div) D-T	14.99
Success w/Fractions (Add/Subt) D-T	14.99
Success w/Fractions (Mult/Div) D-T	14.99
Ducks Ahoy	23.99
Ernie's Magic Shapes	23.99
Murder by the Dozen	23.99
Peanut Butter Panic	24.99

Access Software

Neutral Zone D-T	20.95
Spritemaster D-T	20.95
Beachhead D-T	20.95
Master Composer-D	23.95

Commodore Software

Assembler D	27.95
Easy Finance I, II, III, IV-D	19.95
Easy Calc-D	64.95
Easy Script-D	39.95
Easy Spell-D	19.95
Logo-D	49.95
The Manager-D	37.95
General Ledger	37.95
Accts. Rec.-D	37.95
Accts. Pay-D	37.95
Magic Desk-D	52.95
Silent Butler	24.95
Sky Travel	27.00

SPECIAL HOLIDAY HARDWARE PACKAGES FOR COMMODORE USERS

WOW

Commodore 64, Commodore 1541 Disc Drive, Commodore 803 Printer OR Commodore 64, Commodore 1541 Disc Drive, Color Monitor

YOUR CHOICE **449.95**

COMPACT DISCS

CCA Now Has An Exclusive Line of Compact Disc Players

AOC
Emerson
Sherwood
Magnavox
and Many More

Priced From **159.95**

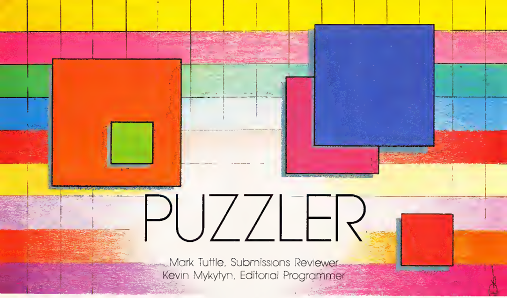
We Have a Huge Selection of Compact Discs - CBS, WEA, Delos, Telarc, Denon, Second Hearing, Mobile Fidelity and Much More!

Call For Our Low, Low Prices

COMPUTER Centers of America Mail Order Ltd.

Call Toll Free:

1-800-631-1003
1-800-221-2760 (in NYS)
1-800-548-0009



PUZZLER

Mark Tuttle, Submissions Reviewer
Kevin Mykytyn, Editorial Programmer

Here's a game that tests your skill in pattern-matching and visualization. It runs on the Commodore 64; unexpanded VIC-20; Plus/4 and 16; IBM PC with color/graphics adapter and BASICA; PCjr with Cartridge BASIC; Apple II-series computers; TI-99/4A with Extended BASIC; and Atari 400/800, XL, and XE computers with at least 16K RAM for tape or 24K for disk. The Commodore 64 and Atari versions also require a joystick.

How good are you at recognizing patterns? Many intelligence tests measure this important conceptual skill. "Puzzler" challenges your ability to find matching patterns in a background of similar shapes. It displays two puzzle grids composed of multicolored blocks (see photos). Both grids contain exactly the same blocks, but those in the left grid have been scrambled. Your job is to rearrange the blocks in the left puzzle grid until they match those on the right. You must solve the puzzle before time runs out.

Because all versions of Puzzler are similar, we've printed general game instructions followed by specific notes for each computer. Read the general instructions as well as the section for your machine, then type in the program listed for your computer. Don't forget to save a copy of the game before you run it.

Puzzle Building

Puzzler begins by letting you choose the size of the puzzle grid. Enter values for the number of rows and columns in the grid. The maximum puzzle size differs among the various versions. Of course, larger puzzles are more difficult to solve than small ones. Next, enter the number of colors the puzzle will use. Two-color puzzles are the easiest. The maximum number of colors depends on which version you're playing. The more colors you choose, the harder your job becomes.

Puzzler then spends a short time building the two grids. Since the blocks are arranged at random, each new puzzle is different from the last. While you try to solve the puzzle, the computer keeps track of the time and alerts you when the puzzle is solved or time runs out. The time limit depends on the size of the puzzle.

Puzzler allows three different operations. You can move within the puzzle grid from one block to another, pick up a block and move it to a new position, or rotate a block in its current position. Use the cursor keys (or joystick in some versions) to move around in the grid. Your position is indicated by a colored cursor (or index arrows in some versions). To pick up a block, press RETURN (or the joystick button) once. The cursor or arrow changes color to show that you're carrying the piece. Then move to the position where you want to place the block, and press RETURN (or the button) once. The block in the current position trades positions with the block you're carrying.

Each block consists of four colored squares. To rotate a block in its current position, press RETURN (or the joystick button) twice. The block rotates 90 degrees. You may rotate a block as many times as you want.

Continue moving and rotating blocks until both puzzle grids match. Every block must match in color and be turned in the right direction.

Commodore 64 Version

Plug a joystick into port 2. The puzzle may contain as many as seven rows and columns, and up to 16 different colors. The box-shaped cursor shows your position on the puzzle grid. Press the joystick button twice without moving the joystick to rotate the block under the cursor. Press the button once to pick up the piece under the cursor: The cursor changes color to show that you're carrying the block. Now you may move to any other place in the grid. When you find the spot

SCENERY DISKS *for* FLIGHT SIMULATOR II

SubLOGIC Scenery Disks expand the potential flying environment of Flight Simulator II and Microsoft Flight Simulator.

Twelve separate Scenery Disks cover the entire continental United States. Each disk covers a geographical region of the country, and includes the major airports, radio-nav aids, cities, highways, rivers, and lakes located in that region. Enough detail is available on each disk for either visual or instrument cross-country navigation.

A STAR Scenery Disk (available 4th quarter 1985) covers a smaller area with a relatively dense amount of scenery. STAR Scenery Disks are primarily intended for visual flight sight-seeing. They include buildings and landmarks, as well as detailed renditions of all major airports in the area.

Individual Scenery Disk price: $19.95
Western U.S. (Disks 1-6): $99.95
– Coming Soon –
Eastern U.S. (Disks 7-12): $99.95

See Your Dealer . . .
or write or call for more information. For direct orders please specify which disk or disk-set you want. Enclose the correct amount plus $2.00 for shipping ($6.00 for the six-disk set) and specify UPS or first class mail delivery. Visa, MasterCard, American Express, and Diners Club cards accepted.

Commodore 64 is a trademark of Commodore Electronics Ltd.
IBM is a registered trademark of International Business Machines Corp.

For the Commodore 64™ and IBM PC®

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

Order Line: (800) 637-4983
(except in Illinois, Alaska, and Hawaii)

you want, press the button again to set down the block. It changes places with the block in that position.

VIC-20 Version

VIC-20 Puzzler is played with the cursor keys. The maximum puzzle size is four columns by six rows, with up to six colors. Your position in the grid is shown by two index arrows, normally colored black. Press RETURN twice to rotate a block. Press RETURN once to pick up a block, then move it with the cursor keys and press RETURN to put it down. The arrows turn blue when RETURN is pressed once, and red when it is pressed a second time.

Plus/4 And 16 Version

Puzzler for the Commodore Plus/4 and 16 permits up to seven rows and columns and seven different colors. It is played exactly like the VIC-20 version.

Atari Version

Plug a joystick into port 1. Atari Puzzler lets you build puzzles with as many as eight rows and columns and up to four different colors. Manipulate the joystick as explained in the Commodore 64 instructions.

IBM Version

IBM Puzzler allows grids as large as seven rows and columns with up to seven different colors. Index arrows indicate your position in the grid, as explained in the VIC-20 instructions. Use the cursor keys to move within the grid. Press Enter to move or rotate a block.

TI-99/4A Version

You have the option of playing with either a joystick or keyboard controls. Puzzles can be as large as six rows and six columns with as many as six different colors. The box-shaped cursor shows your position in the puzzle grid and changes colors to indicate when you're carrying a block. When using the keyboard, make sure the Alpha Lock key is down. Move the cursor with the arrow keys and press Enter to rotate or move a block.

Apple Version

Puzzler runs on any Apple II-series computer with either DOS 3.3 or ProDOS. Press the space bar to move or rotate a block, and press I, J, K, and L to move up, left, down, and right, respectively. Your position in the grid is indicated by small white highlights in the corners of the block.

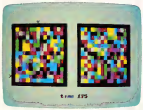
Program 1: Commodore 64 Puzzler

Version by John Krause, Assistant Technical Editor

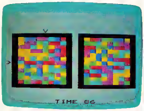
For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
100 GOSUB460                   :rem 171
110 IFT<TI/60THENPOKE53269,0:G
    OTO790                     :rem 185
120 A=INT(T-TI/60+.5):B=INT(A/
    60)                         :rem 89
130 PRINT"{HOME}{3 DOWN}"SPC(1
    7)B"{LEFT}:";              :rem 226
140 Z$=STR$(A-60*B):Z$=RIGHT$(
    Z$,LEN(Z$)-1):IFLEN(Z$)=1T
    HENPRINT"0";               :rem 236
150 PRINTZ$                    :rem 161
160 A=NOTPEEK(56320)           :rem 124
170 R=R+SGN((AAND2)-(AAND1))
                                :rem 55
180 C=C+SGN((AAND8)-(AAND4))
                                :rem 35
190 IFR<0THENR=0               :rem 213
200 IFR>=R3THENR=R3-1           :rem 20
210 IFC<0THENC=0               :rem 176
220 IFC>=C3THENC=C3-1          :rem 218
230 POKE53248,CS+16*C:POKE5324
    9,RS+16*R                  :rem 218
240 IF(AAND16)=0THEN110:rem 31
250 IFF=0THENF=1:POKE53287,14:
    RR=R:CC=C:WAIT56320,16:GOT
    O110                       :rem 115
260 F=0:IFRR=RANDCC=CTHENGOSUB
    300:GOTO280                :rem 111
270 GOSUB330                   :rem 175
280 POKE53287,15:WAIT56320,16:
    IFA$<>B$THEN110             :rem 53
290 GOTO800                    :rem 108
300 B=C1+80*R+C+C+41:GOSUB420
                                :rem 81
310 POKEB,PEEK(A):POKEB+1,PEEK
    (A+1)                       :rem 46
320 POKEB+40,PEEK(A+NC):POKEB+
    41,PEEK(A+NC+1):RETURN
                               :rem 132
330 GOSUB360:B=C1+80*R+2*C+41:
    GOSUB310                   :rem 147
340 B=C1+80*RR+2*CC+41:A=AA:GO
    TO310                      :rem 201
350 REM *** MOVE                :rem 49
360 AA=SS+2*NC*RR+2*CC:A=SS+2*
    NC*R+C+C                   :rem 191
370 D=PEEK(A):POKEA,PEEK(AA):P
    OKEAA,D                    :rem 251
380 D=PEEK(A+1):POKEA+1,PEEK(A
    A+1):POKEAA+1,D            :rem 108
390 D=PEEK(A+NC):POKEA+NC,PEEK
    (AA+NC):POKEAA+NC,D
                               :rem 237
400 D=PEEK(A+NC+1):POKEA+NC+1,
    PEEK(AA+NC+1):POKEAA+NC+1,
    D:RETURN                   :rem 111
410 REM *** ROTATE             :rem 198
420 A=SS+2*NC*R+C+C             :rem 42
430 D=PEEK(A):POKEA,PEEK(A+NC)
                                :rem 24
440 POKEA+NC,PEEK(A+NC+1)
                               :rem 191
450 POKEA+NC+1,PEEK(A+1):POKEA
    +1,D:RETURN                :rem 240
460 POKE53269,0:A$="":POKE5328
    0,6:POKE53281,6            :rem 233
470 PRINT"{CLR}"CHR$(14)SPC(16
    )"{2 DOWN}{WHT}PUZZLER":PR
    INTSPC(16)"[7 T]"         :rem 153
480 FORT=54272TO54295:POKET,0:
    NEXT:POKE54296,15           :rem 91
490 INPUT"{HOME}{7 DOWN}NUMBER
     OF ROWS (2-7)";R3:rem 203
500 IFR3<2ORR3>7THEN490
                               :rem 126
510 INPUT"{HOME}{10 DOWN}NUMBE
    R OF COLUMNS (2-7)";C3
                               :rem 190
520 IFC3<2ORC3>7THEN510:rem 91
530 INPUT"{HOME}{13 DOWN}NUMBE
    R OF COLORS (2-14)";CO
                               :rem 238
540 IFCO<2ORCO>14THEN530
                               :rem 197
550 PRINT"{2 DOWN}PLEASE WAIT
    {SPACE}..."                :rem 134
560 S1=1473-40*R3-C3:C1=S1+542
    72:S2=S1+20:C2=C1+20:NR=2*
    R3:NC=2*C3                 :rem 120
570 FORA=1TONR*NC:A$=A$+CHR$(R
    ND(1)*CO):NEXT:B$=A$
                               :rem 203
580 A=256*PEEK(46)+PEEK(45)
                               :rem 204
590 SS=256*PEEK(A+4)+PEEK(A+3)
                               :rem 158
600 FORR=0TOR3-1:FORC=0TOC3-1:
    B=INT(RND(1)*4)            :rem 195
610 IFBTHENGOSUB420:B=B-1:GOTO
    610                         :rem 16
620 NEXT:NEXT                   :rem 80
630 FORR=0TOR3-1:FORC=0TOC3-1
                                :rem 13
640 RR=INT(RND(1)*R3):CC=INT(R
    ND(1)*C3):GOSUB360:NEXT:NE
    XT                          :rem 80
650 PRINT"{CLR}"SPC(17)"{DOWN}
    PUZZLER                    :rem 141
660 FORA=1TONR:FORB=1TONC:POKE
    C1+40*A+B,PEEK(SS+E)
                               :rem 118
670 POKES1+40*A+B,160:E=E+1:NE
    XT:NEXT                    :rem 201
680 FORA=1TONR:FORB=1TONC:POKE
    C2+40*A+B,ASC(MID$(B$,G+1)
    )                          :rem 153
690 POKES2+40*A+B,160:G=G+1:NE
    XT:NEXT                    :rem 208
700 POKE2040,14:POKE53287,15:P
    OKE53277,1:POKE53271,1
                               :rem 183
710 FORA=896TO924:READB:POKEA,
    B:NEXT                      :rem 15
720 FORA=925TO958:POKEA,0:NEXT
                               :rem 102
730 RS=144-4*NR:CS=102-4*NC:R=
    0:C=0                      :rem 223
740 T=NR*NC*3:POKE53269,1:TI$=
    "000000":RETURN            :rem 105
750 DATA255,192,0,128,64,0,128
    ,64,0                      :rem 232
760 DATA128,64,0,128,64,0,128,
    64,0                       :rem 182
770 DATA128,64,0,128,64,0,128,
    64,0                       :rem 183
780 DATA255,192                 :rem 29
790 Z1=50:Z2=10:Z3=-2:GOSUB830
    :PRINT"{HOME}{DOWN}"SPC(15
    )"{YEL}TIME'S UP":GOTO820
                               :rem 114
800 Z1=10:Z2=50:Z3=2:GOSUB830
                               :rem 180
810 PRINT"{HOME}{DOWN}"SPC(13)
    "{YEL}YOU SOLVED IT!"
                                :rem 19
```

```
820 PRINTTAB(8)"{DOWN}FIRE BUT
    TON TO PLAY AGAIN{HOME}":W
    AIT56320,16,16:RUN:rem 238
830 A=15:D=107:POKE53248+21,0:
    POKE54277,A:POKE54284,A:PO
    KE54291,A:POKE5427B,D
                        :rem 165
840 POKE54285,D:POKE54292,D:PO
    KE54286,50:POKE542B7,40:PO
    KE54276,33           :rem 43
B50 POKE54283,33:POKE54290,33:
    FORF1=Z1TOZ2STEPZ3:POKE542
    73,F1:POKE54287,F1:rem 226
860 FORF2=30TO1STEP-5:POKE5428
    0,F2:POKE53280,F2:NEXTF2,F
    1                   :rem 194
870 POKE54276,32:POKE54283,32:
    POKE54290,32:RETURN
                        :rem 127
880 POKE54277,26:POKE54276,23:
    POKE54273,30:RETURN
                        :rem 133
```



"Commodore 64 Puzzler" permits large puzzles with up to 16 different colors.



Index arrows indicate your position in "VIC-20 Puzzler."

Program 2: VIC-20 Puzzler

Version by Kevin Mykytyn, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!

```
10 X$="{RVS} {OFF}":CO(0)=0:CO
   (1)=6:CO(2)=2:DN$="{HOME}
   {22 DOWN}"             :rem 53
20 PRINT"{RED}{CLR}{4 DOWN}"SP
   C(8)"PUZZLER":PRINT"
   {4 DOWN}{BLK}{4 SPACES}ENTE
   R GRID SIZE":LN=2:HN=4
                          :rem 87
30 PRINT"{2 DOWN}{4 SPACES}COL
   UMNS? (2-4)":GOSUB370:COL=A
   :HN=6                 :rem 199
40 PRINT"{2 DOWN}{4 SPACES}ROW
   S?{4 SPACES}(2-6)":GOSUB370
   :ROW=A                :rem 203
50 PRINT"{2 DOWN}HOW MANY COLO
   RS? (2-6)":GOSUB370:CR=A
                          :rem 19
70 PRINT"{CLR} {BLK} CONSTRUCT
   ING PUZZLE":PRINTDN$SPC(5)"
   {BLU}PLEASE WAIT{BLK}";
                         :rem 238
80 PRINTLEFT$(DN$,11-ROW):GOSU
   B390                   :rem 84
90 FORI=1TOROW*2:PRINTLEFT$(DN
   $,I+12-ROW);:PRINTSPC(5-COL
   )X$;:PRINTSPC(COL*2)X$;
                          :rem 98
95 PRINTSPC(9-2*COL)X$;:PRINTS
   PC(2*COL)X$;:NEXT:IFCOL<>4T
   HENPRINT                :rem 2
100 GOSUB390:A$="":FORA=1TOROW
    *COL*4:A$=A$+CHR$(INT(RND(
    1)*CR)+2):NEXTA:B$=A$
                          :rem 22
110 FORA=1TOROW*COL:Q=(A-1)*4+
    1:Q2=INT(RND(1)*ROW*COL)*4
    +1:GOSUB400:NEXT  :rem 132
120 FORA=1TOROW*COL:R=RND(1)*4
    :Q=(A-1)*4+1:GOSUB410:NEXT
    :IFA$=B$THEN110      :rem 37
130 FORA=1TOROW*COL:Q=(A-1)*4+
    1:T$=B$:XBAS=17-COL:YBAS=1
    3-ROW:GOSUB440:XBAS=6-COL
                          :rem 19
135 T$=A$:GOSUB440:NEXT
                         :rem 126
140 PRINT"{HOME}{OFF}
    {21 SPACES}":PRINTDN$"
    {BLK}{8 SPACES}TIME
    {4 SPACES}";        :rem 198
150 A=1:PB=1:OA=1:FL=0:TM=ROW*
    COL*2.5+30            :rem 85
160 IFFL=1THENQ2=(OA-1)*4+1:FL
    =2                    :rem 73
170 ZZ=A:A=OA:GOSUB460:A=ZZ:XP
    =XBAS-2:YP=YBAS+Y1*2:GOSUB
    600:PRINT" ";       :rem 116
175 YP=YBAS-2:XP=XBAS+X1*2:GOS
    UB600:PRINT" ";:GOSUB460
                          :rem 88
180 POKE646,CO(FL):YP=YBAS+Y1*
    2:XP=XBAS-2:GOSUB600:PRINT
    ">";:XP=XBAS+X1*2:YP=YBAS-
    2                    :rem 244
190 GOSUB600:POKE646,CO(FL):PR
    INT"V";:OA=A:MAX=ROW*COL:G
    ETK$                  :rem 36
200 IFK$="{UP}"THENA=A-COL:GOT
    O240                 :rem 164
210 IFK$="{LEFT}"THENA=A-1:GOT
    O240                   :rem 4
220 IFK$="{RIGHT}"THENA=A+1:GO
    TO240                :rem 131
230 IFK$="{DOWN}"THENA=A+COL:G
    OTO240                :rem 37
235 GOTO250              :rem 106
240 IFA>MAXORA<1THENA=OA
                          :rem 22
250 IFK$=OK$THEN290       :rem 90
260 OK$=K$:IFK$=CHR$(13)ANDFL=
    0THENFL=1:GOTO290     :rem 60
270 IFK$=CHR$(13)ANDFL=1THENQ=
    (A-1)*4+1:R=1:GOSUB410:T$=
    A$:GOSUB440:FL=0:GOTO290
                         :rem 250
2B0 IFK$=CHR$(13)ANDFL=2THENGO
    SUB470:FL=0          :rem 226
290 PRINTDN$SPC(12)INT(TM)"
    {LEFT} ";:TM=TM-.0B
                         :rem 237
300 IFTM<0THENPRINTDN$"{BLU}
    {OFF}{7 SPACES}TIME'S UP
    {3 SPACES}";:Z1=255:Z2=150
    :GOTO330              :rem 95
310 IFA$=B$THENPRINTDN$"{BLU}
    {OFF}{3 SPACES}YOU SOLVED
    {SPACE}IT{3 SPACES}";:Z1=1
    50:Z2=255:GOTO330 :rem 116
320 IFA<>OATHEN160        :rem 53
322 IFK$=CHR$(13)THEN170
                          :rem 79
325 GOTO190              :rem 109
330 GOSUB360:PRINTDN$"
    {4 SPACES}PRESS ANY KEY";
                           :rem 9
340 POKE198,0:WAIT198,1:RUN
                          :rem 97
360 POKE36878,15:FORA=Z1TOZ2ST
    EP2*SGN(Z2-Z1):POKE36875,A
    :POKE36874,A-5       :rem 255
365 POKE36879,(PEEK(36879)AND2
    48)ORRND(1)*8:NEXTA:FORT=1
    5TO0STEP-1:POKE36878,T:NEX
    T                    :rem 109
367 POKE36879,27:RETURN
                         :rem 143
370 Z=RND(1):GETK$:A=VAL(K$):I
    FA<LNORA>HNTHEN370:rem 218
380 RETURN               :rem 123
390 PRINTSPC(5-COL);:FORI=1TO(
    COL+1)*2:PRINTX$;:NEXT:PRI
    NTSPC(9-2*COL);       :rem 13
395 FORI=1TO(COL+1)*2:PRINTX$;
    :NEXT:RETURN         :rem 218
400 T$=A$:GOSUB500:FORZ=0TO3:T
    =PEEK(Q+Z+BP):POKEQ+Z+BP,P
    EEK(Q2+Z+BP)         :rem 242
405 POKEQ2+Z+BP,T:NEXT:A$=T$:R
    ETURN                :rem 244
410 IFR=0THENRETURN      :rem 242
420 T$=A$:GOSUB500:FORX=1TOR:T
    =PEEK(BP+Q):POKEBP+Q,PEEK(
    BP+Q+2):POKEBP+Q+2,PEEK(BP
    +Q+3)                :rem 144
430 POKEBP+Q+3,PEEK(BP+Q+1):PO
    KEBP+Q+1,T:NEXT:A$=T$:RETU
    RN                   :rem 120
440 GOSUB460:XP=XBAS+X1*2:YP=Y
    BAS+Y1*2:GOSUB600:FORT=0TO
    3:IFT=2THENYP=YP+1:GOSUB60
    0                    :rem 109
450 POKE646,ASC(MID$(T$,Q+T,1)
    ):PRINTX$;:NEXT:RETURN
                         :rem 237
460 Z=A-1:Y1=INT(Z/COL):X1=Z-Y
    1*COL:RETURN         :rem 167
470 Q=(A-1)*4+1:GOSUB400:T$=A$
    :GOSUB440:ZZ=A:AA=Q:Q=Q2:A
    =(Q2-1)/4+1:T$=A$:GOSUB440
    :A=ZZ                :rem 131
480 Q=AA:RETURN          :rem 198
500 T$=T$:BP=PEEK(51)+256*PEEK
    (52)-1:RETURN        :rem 23B
600 PRINTLEFT$(DN$,YP)SPC(XP);
    :RETURN              :rem 130
```

Program 3: Puzzler For Commodore Plus/4 And 16

Version by Patrick Parrish, Programming Supervisor

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!

```
10 X$="{RVS} {OFF}":CO(0)=1:CO
   (1)=7:CO(2)=3:DN$="{HOME}
   {22 DOWN}":COLOR0,2:COLOR4,
   2
20 PRINT"{7}{CLR}{6 DOWN}"SPC(
   16)"PUZZLER":PRINT"{3 DOWN}
```

THE SHADOW KNOWS

99.9% EFFECTIVE

Shadow is a new and revolutionary way to back up your most protected software. It encompasses all the latest advances in software, as well as a highly sophisticated piece of hardware. This is absolutely **the best** utility available today. You may not even need updates, but if you do, the Shadow's unique design allows for updates to be available for a nominal fee.

By the time you place your order we expect the Shadow to back up 100% — that's right, 100% — of **all** software available for the C-64.

Order by phone 24 hrs./7 days or send cashier's check/money order payable to Megasoft. Visa, MasterCard include card # and exp. date. Add $3.50 shipping/handling for continental U.S., $5.50 for UPS air. CODs add $7.50, Canada add $10.00 Other foreign orders add $15.00 and remit certified U.S. funds only. Distributors invited and supported.

MegaSoft LTD.

INTRODUCTORY OFFER $89.95

P.O. Box 1080 Battle Ground, Washington 98604
Phone (206) [illegible] BBS 687-5205 After-Hours Computer-to-Computer Ordering

Lyco Computer Marketing & Consultants

SAVE ON THESE IN STOCK PRINTERS

SG-10 ...... $212.00

AXIOM
- GP550AT (Atari) 222
- GP550CD (C-64) 222
- GP700AT (Atari) 439
- GP700AP (Apple) 439
- ELITE5CD (C-64) 229

C. ITOH
- 8510Ap+ 299
- 1550P 437
- 1550BP 437
- 1550EP 437
- 1550BCD 469
- F10-40 859
- F10-55 1039
- 7500AP 204
- 7500AR 244

TOSHIBA
- P1340 557
- P351 1288

CARDCO
- 32K BUFFER (C-64) 59

CORONA
- LP300 Laser Printer 2686
- 200361 Toner Cartridge 89

EPSON
- FX85 (New) 333
- LX80 212
- FX185 (New) 464
- LX90 (New) 226
- SQ2000 (New) 1555
- JX80 467
- Homewriter 10 193
- CR-20-Atari 153
- CR-220-C64 153
- DX-10 (New) 207
- DX-20 (New) 297
- HS-80 (New) 268
- LQ1500P 975
- LQ1500S 1039
- RX-100 356
- FX-100+ CALL

CITIZEN
- MSP-10 287
- MSP-15 447
- MSP-20 447
- MSP-25 563
- Sheetfdr (10/20) 189
- Sheetfdr (15/25) 199

OKIDATA
- Okimate 10 CALL
- Okimate 20 CALL
- 182 214
- 192 348
- 193 563
- 92P 349
- 93P 565
- 84P 645
- 92 Imagewriter 349

*IBM versions also

BROTHER
- HR-15XL-P 359
- HR-15XL-S 359
- HR-35P 839
- HR-35S 839
- 2024L-P 949
- M1009-P 189

JUKI
- Juki 6100 347
- RS232 Serial Board 55
- 6100 Tractor 119
- 6100 Sheet Feeder 209
- Juki 6300 757

LEGEND
- 880 196
- 1080 222
- 1380 262
- 1385 296

DIGITAL DEVICES
- 16K BUFFER 75
- 32K BUFFER 89
- 64K BUFFER 125

DIABLO
- Q25 549
- 630API 1599
- 630ECS 1759
- Q801F 2395
- P32CQ1 699
- P38 1749
- C150 999

1091 .... $233.00

PANASONIC
- 1091 233
- 1090 183
- 1092 373
- 1093 426
- 3151 Letter 426
- 4K Buffer 65

SILVER REED
- EXP400 249
- EXP500 295
- EXP550 399
- EXP770 749

STAR MICRONICS
- SG-10 212
- SG-15 373
- SD-10 336
- SD-15 442
- SR-10 483
- SR-15 583
- Powertype 303
- SB-10 CALL
- SG-10C-64 (NEW) CALL

★ PRINTER ★ INTERFACING

AXIOM
- ATB46 (Atari) 65
- Parallax (C-64) 65

MICROBITS
- MPP-1150 (Atari) 54
- MPP-1150XL (Atari) 59
- MicroPrint (Atari) 39
- 64K RAM Board (600xl) 65

DIGITAL DEVICES
- ApeFace XLP (Atari) 49
- ApeFace 12XLP (1200xl) 49
- U-Print A (Atari) 54
- U-Print C (C-64) 49
- U-Print A16K Buffer 74
- U-Print A64K Buffer 99
- U-Print AP16K (Apple) 99
- U-Print AP64K (Apple) 139
- U-CALL RS232 (Atari) 37

MICROTEK
- Dumpling GX (Apple) 59
- Dumpling 16K (Apple) 89
- RV-611C (Apple) 49

TYMAC
- Connection (C-64) 55
- Tackler (Apple) 49
- PPC-100 (Apple) 39

ORANGE MICRO
- GRAPPLER + (Apple) B5
- Grappler 16K (Appla) 149
- ORANGE (Apple) 59
- Grappler CD (C-64) 79

CARDCO
- G-Wiz (C-64) 54
- C/? + G (C-64) 45
- C/?PS (C-64) 49
- C/?B (C-64) 39
- C/?AT (Atari) 49
- C/?AP (Apple) 49
- C/?IBM (IBM) 29

PRINTING PAPER

3000 SHEETS FANFOLQ	$42 75
1000 SHEETS FANFOLQ	$19 75
1000 SHEET LETTER	$21 95
200 SHEETS LETTER	$8 99
150 RAG STATIONARY	$10 99
MAILING LABELS (1in)	$9 95
14 x 11 1000 FANFOLD	$24 75

WICO Joysticks
- 15-9714 Bat Handle 16.75
- 50-2030 Boss 11 99
- 50-2002 Super 3-Way 19 99
- 50-0110 IBM-Analog 29 95
- 50-1030 Appla-Analog 29 95
- 72-4545 Trackball 29 95

INNOVATIVE CONCEPTS
- Flip-n-File 10 3.50
- Flip-n-File 15 B 25
- Flip-n-File 25 Lock 17 95
- Flip-n-File 50 17 25
- Flip-n-File 50 Lock 22 95
- Flip-n-File Rom 17 25

DRIVES

INDUS
- *GT ATARI* 199
- GT COMMODORE 199

MSD
- SD1 Drive (C-64) 229
- SD2 Drive (C-64) 469

DISKETTES

SKC (Box 10)

(Qty)	10	100	500
5¼ SSDD	13 75	12 99	11 99
5¼ DSDD	15 75	14 99	14 25

ELEPHANT (Box 10)

5¼ SSSD	13 50	12 99	11 99
5¼ SSDD	14 99	13 99	12 99
5¼ DSDD	16 99	15 99	14 99
Premium SSDD	15 95	14 95	13 95
Premium DSDD	17 95	16 95	15 95

MAXELL (Box 10)

5¼ MD 1	16 75	15 75	14 75
5¼ MD 2	22 99	21 99	20 99

VERBATIM (DataLife) (Box 10)

5¼ SSDD	19 99	18 99	17 99
5¼ DSDD	24 99	23 99	22 99

NO LABEL DISKETTES
- NL 5¼" SSDD 10 99 (Box 10)
- NL 5¼" DSDD 15 99 (Box 10)

*Free Diskette Writer Pen!
*Free Storage Case!
*Lifetime Warranty!

IBM-PC SOFT-WARE

*ASHTON-TATE
- Framework 3B5 00
- d Base II 2B9 00
- d Base III 385 00
- d Base II upgrade 135 00

PARADISE
- Multi-Display Card 2B9
- Modular Graphics Card 275
- Five-Oak 155

*LOTUS
- Lotus 1-2-3 309 00
- Symphony 439 00

IBM-PC COMPATABLE

LEADING EDGE
- PC Compatable Call

ZENITH
- Z-150 Call

TECHMAR
- First Mate-64K 249
- Captain-64K 199
- Second Mate 209
- Captain JR-12BK 309
- Graphics Master 465

OMEGA
- 10 Meg Bernoulli Box (IBM) 1899
- 5 Meg Bernoulli Box (MAC) 1499

TALLGRASS TECHNOLOGIES
- 25mb drive CALL
- 35mb drive CALL
- 50mb drive CALL
- 80mb drive CALL

AST
- Six Pack Plus 64K CALL
- Mega Plus II 64K 269
- Memory 128K (mega) 199
- Memory 256K (Mega) 335
- Mono-Graph Plus 339
- Advantage-AT-12BK 3B9

*QUADRAM
- Quad Jr Exp Chassis 519 00
- Ouad Jr Exp Memory 209 00
- Quad Memory Jr 209 00

*LEADING EDGE
- Nutshell 69 95
- LEWP Basic 65 00
- LEWP Merge Print 99 00
- LE Spell Correction 169 00

TOLL FREE 1-800-233-8760

TO ORDER

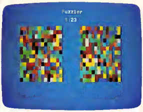
CALL TOLL FREE
800-233-8760
Customer Service 1-717-327-1825

or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA
17740

RISK FREE POLICY

In-stock items shipped within 24 hours of order. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. Volume discounts available. PA residents add sales tax. APO, FPO, and international orders add $5.00 plus 3% for priority mail service. Advertised prices show 4% discount for cash, add 4% for MasterCard or Visa. Personal checks require 4 weeks' clearance before shipping. Ask about UPS Blue and Red label shipping. All merchandise carried under manufacturer's warranty. Free catalog with order. All items subject to change without notice.

```
     {BLK}"SPC(12)"ENTER GRID SI
     ZE:":LN=2:HN=7
30 PRINT"{DOWN}"SPC(12)"COLUMN
     S? (2-7)":GOSUB430:COL=A
40 PRINT"{DOWN}"SPC(12)"ROWS?
     {4 SPACES}(2-7)":GOSUB430:R
     OW=A
50 PRINT"{DOWN}"SPC(10)"HOW MA
     NY COLORS? (2-7)":GOSUB430:
     CR=A
60 PRINT"{CLR}"SPC(10)"CONSTRU
     CTING PUZZLE":PRINTDN$SPC(1
     4)"E7}PLEASE WAIT{BLK}";
70 PRINTLEFT$(DN$,11-ROW):GOSU
     B450
80 FORI=1TOROW*2:PRINTLEFT$(DN
     $,I+12-ROW);:PRINTSPC(9-COL
     )X$;:PRINTSPC(COL*2)X$;
90 PRINTSPC(17-2*COL)X$;:PRINT
     SPC(2*COL)X$;:NEXT:PRINT
100 GOSUB450:A$="":FORA=1TOROW
      *COL*4:A$=A$+CHR$(INT(RND(
      Ø)*CR)+2):NEXTA:B$=A$
110 FORA=1TOROW*COL:Q=(A-1)*4+
      1:Q2=INT(RND(1)*ROW*COL)*4
      +1:GOSUB470:NEXT
120 FORA=1TOROW*COL:R=RND(1)*4
      :Q=(A-1)*4+1:GOSUB490:NEXT
      :IFA$=B$THEN110
130 FORA=1TOROW*COL:Q=(A-1)*4+
      1:T$=B$:XBAS=29-COL:YBAS=1
      3-ROW:GOSUB520:XBAS=10-COL
140 T$=A$:GOSUB520:NEXT
150 PRINT"{HOME}{OFF}"SPC(10)"
      {19 SPACES}":PRINTDN$SPC(1
      4)"{2 SPACES}E7}TIME
      {6 SPACES}";
160 A=1:PB=1:OA=1:FL=Ø:TM=ROW*
      COL*3+30
170 IFFL=1THENQ2=(OA-1)*4+1:FL
      =2
180 ZZ=A:A=OA:GOSUB540:A=ZZ:XP
      =XBAS-2:YP=YBAS+Y1*2:GOSUB
      580:PRINT" ";
190 YP=YBAS-2:XP=XBAS+X1*2:GOS
      UB580:PRINT" ";:GOSUB540
200 COLOR1,CO(FL),4:YP=YBAS+Y1
      *2:XP=XBAS-2:GOSUB580:PRIN
      T">";:XP=XBAS+X1*2:YP=YBAS
      -2
210 GOSUB580:COLOR1,CO(FL),4:P
      RINT"V";:OA=A:MAX=ROW*COL:
      GETK$
220 IFK$="{UP}"THENA=A-COL:GOT
      O270
230 IFK$="{LEFT}"THENA=A-1:GOT
      O270
240 IFK$="{RIGHT}"THENA=A+1:GO
      TO270
250 IFK$="{DOWN}"THENA=A+COL:G
      OTO270
260 GOTO280
270 IFA>MAXORA<1THENA=OA
280 IFK$=OK$THEN320
290 OK$=K$:IFK$=CHR$(13)ANDFL=
      ØTHENFL=1:GOTO320
300 IFK$=CHR$(13)ANDFL=1THENQ=
      (A-1)*4+1:R=1:GOSUB490:T$=
      A$:GOSUB520:FL=Ø:GOTO320
310 IFK$=CHR$(13)ANDFL=2THENGO
      SUB550:FL=Ø
320 PRINTDN$"{BLK}"SPC(20)INT(
      TM)"{LEFT} ";:TM=TM-.08
330 IFTM<ØTHENPRINTDN$SPC(10)"
      E7}{OFF}{5 SPACES}TIME'S U
      P{3 SPACES}";:Z1=1023:Z2=Ø
      :GOTO380
340 IFA$=B$THENPRINTDN$SPC(10)
      "E7}{OFF}{3 SPACES}YOU SOL
      VED IT!{2 SPACES}";:Z1=Ø:Z
      2=1023:GOTO380
350 IFA<>OATHEN170
```



"Puzzler" for the Commodore Plus/4 and 16 uses keyboard controls.

```
360 IFK$=CHR$(13)THEN180
370 GOTO210
380 GOSUB400:PRINTDN$SPC(10)"
      {3 SPACES}PRESS ANY KEY";
390 POKE239,Ø:WAIT239,1:RUN
400 VOL 8:FORA=Z1TOZ2STEP10*SG
      N(Z2-Z1):SOUND 1,A,2
410 COLORØ,RND(1)*15+1:NEXTA:F
      ORT=8TOØSTEP-1:VOL T:NEXT
420 COLORØ,2:COLOR4,2:RETURN
430 Z=RND(1):GETK$:A=VAL(K$):I
      FA<LNORA>HNTHEN430
440 RETURN
450 PRINTSPC(9-COL);:FORI=1TO(
      COL+1)*2:PRINTX$;:NEXT:PRI
      NTSPC(17-2*COL);
460 FORI=1TO(COL+1)*2:PRINTX$;
      :NEXT:RETURN
470 T$=A$:GOSUB570:FORZ=ØTO3:T
      =PEEK(Q+Z+BP):POKEQ+Z+BP,P
      EEK(Q2+Z+BP)
480 POKEQ2+Z+BP,T:NEXT:A$=T$:R
      ETURN
490 IFR=ØTHENRETURN
500 T$=A$:GOSUB570:FORX=1TOR:T
      =PEEK(BP+Q):POKEBP+Q,PEEK(
      BP+Q+2):POKEBP+Q+2,PEEK(BP
      +Q+3)
510 POKEBP+Q+3,PEEK(BP+Q+1):PO
      KEBP+Q+1,T:NEXT:A$=T$:RETU
      RN
520 GOSUB540:XP=XBAS+X1*2:YP=Y
      BAS+Y1*2:GOSUB580:FORT=ØTO
      3:IFT=2THENYP=YP+1:GOSUB58
      Ø
530 P=ASC(MID$(T$,Q+T,1)):COLO
      R1,P+(P=4)*2,P-1-(P=4)*4:P
      RINTX$;:NEXT:RETURN
540 Z=A-1:Y1=INT(Z/COL):X1=Z-Y
      1*COL:RETURN
550 Q=(A-1)*4+1:GOSUB470:T$=A$
      :GOSUB520:ZZ=A:AA=Q:Q=Q2:A
      =(Q2-1)/4+1:T$=A$:GOSUB520
      :A=ZZ
560 Q=AA:RETURN
570 T$=T$:BP=PEEK(51)+256*PEEK
      (52)-1:RETURN
580 PRINTLEFT$(DN$,YP)SPC(XP);
      :RETURN
```

Program 4: Atari Puzzler

Version by Kevin Mykytyn, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!

```
AG 10 OPEN #1,4,12,"K:":POKE
         106,PEEK(106)-8:GRAPH
         ICS Ø:CHBAS=PEEK(106)*
         256:POKE 82,Ø:SOUND Ø,
         Ø,Ø,Ø
CI 20 POKE 752,1:POSITION 14
         ,11:PRINT "PLEASE WAIT
         "
IJ 30 FOR A=Ø TO 1023:POKE C
         HBAS+A,PEEK(57344+A):N
         EXT A:FOR A=CHBAS+8 TO
          CHBAS+39:READ B:POKE
         A,B:NEXT A:GRAPHICS Ø
GJ 40 SPRBAS=PEEK(106)+4:POK
         E 53277,3:POKE 623,1:P
         OKE 704,Ø
MM 50 SPR=SPRBAS*256+512:OY=
         SPR:FOR A=SPR TO SPR+1
         27:POKE A,Ø:NEXT A:POK
         E 53256,1
HD 60 DIM T$(256),A$(256),B$
         (256),T2$(1),R(4),SP(8
         ):FOR A=1 TO 8:READ Z:
         SP(A)=Z:NEXT A
BF 70 GRAPHICS 17:POSITION 7
         ,6:PRINT #6;"puzzler":
         FOR A=1 TO 3:R(A)=32+A
         :NEXT A:R(4)=161
EE 80 POSITION 3,10:PRINT #6
         ;"enter grid size":LN=
         3:HN=8
HG 90 POSITION 3,14:PRINT #6
         ;"COLUMNS ? (3-8)":GOS
         UB 480:COL=A
LK 100 POSITION 3,14:PRINT #
          6;"ROWS{3 SPACES}":GO
          SUB 480:ROW=A
BO 110 POSITION 3,10:PRINT #
          6;"HOW MANY COLORS ?"
          :LN=2:HN=4
IN 120 POSITION 3,14:PRINT #
          6;"{5 SPACES}(2-4)
          {6 SPACES}":GOSUB 480
          :COLR=A
FD 130 GRAPHICS Ø:DL=PEEK(56
          Ø)+256*PEEK(561):POKE
           DL+3,66:FOR I=DL+6 T
          O DL+27:POKE I,4:NEXT
           I
EN 140 POKE I,6:I=I+1:POKE I
          ,65:POKE I+1,Ø:POKE I
          +2,DL/256:POKE 82,Ø
GP 150 POSITION 11,Ø:PRINT "
          CONSTRUCTING PUZZLE":
          POSITION 5,23:PRINT "
          PLEASE WAIT";:POKE 75
          6,CHBAS/256
ED 160 POKE 559,46:POKE 5427
          9,SPRBAS:POSITION Ø,1
          2-ROW:GOSUB 520
IF 170 FOR I=1 TO ROW*2:PRIN
          T :POKE 85,10-COL:PRI
          NT "$";:POKE 85,11+CO
          L:PRINT "$";:POKE 85,
          29-COL:PRINT "$";
BB 180 POKE 85,30+COL:PRINT
          "$";:NEXT I:PRINT :GO
          SUB 520
FD 190 FOR A=1 TO ROW*COL*4:
          A$(A,A)=CHR$(R(INT(RN
          D(1)*COLR+1))):B$(A,A
          )=A$(A,A):NEXT A
MJ 200 FOR A=1 TO ROW*COL:Q=
          (A-1)*4+1:Q2=INT(RND(
          1)*ROW*COL)*4+1:GOSUB
           530:NEXT A
GK 210 FOR A=1 TO ROW*COL:R=
          RND(1)*4:Q=(A-1)*4+1:
          GOSUB 540:NEXT A:IF A
          $=B$ THEN 200
KC 220 FOR A=1 TO ROW*COL:Q=
          (A-1)*4+1:T$=B$:XBAS=
          30-COL:YBAS=13-ROW:GO
          SUB 570:XBAS=11-COL:T
          $=A$:GOSUB 570:NEXT A
KM 230 POSITION 11,Ø:PRINT "
```

SOLO FLIGHT

"The only Flight Simulator that gives you the real feel and thrill of Primary Flight Training with the FUN, Ease of Use and challenge even for the best fighter pilot!"

Major Bill
3,000 Flying Hours

NOW INCLUDES

MAIL PILOT MISSION

Brave the wind, weather, and mechanical failures in an exciting and increasingly challenging race to deliver the mail to 21 different airports from Kansas to Colorado!

INSTRUCTOR PILOT OPTION

On screen guidance from a real instructor pilot to help you make your first Solo!

CESSNA and MICRO PROSE

present the

Great American

LEARN TO FLY A REAL AIRPLANE

Contest

$30 OFF COUPON AND FREE FLYING LESSONS!!!

You get a coupon worth up to $30 towards a Cessna "Discovery Flight" in every flying simulation product box — Plus you can win a complete flight training course from Cessna. Details in every box, or see your local dealer.

A Truly Unique Flying Experience –

Solo Flight is an exciting and sophisticated Primary Flight Training Simulator that makes learning to fly Fun and Easy! Designed by a real fighter pilot, who taught more than 200 Air Force Lieutenants how to fly, Solo Flight takes you from your first taxi to take-off, through your first solo, and all the way up to the challenge of the Mail Pilot Mission!

Solo Flight features full instrumentation, multiple air navigation maps, emergency procedure practice, the unique "in-the-slot" 3-Dimensional flight perspective, and fun and challenge for pilots of all ages!

USA TODAY – SOLO FLIGHT

"... there's a revised pecking order in the air... easier to master (than another famous flight simulator) (Solo Flight)... loses none of the subtlety of flight, even when you've logged dozens of hours."

Marc Randolph

CREATIVE COMPUTING – SOLO FLIGHT

"... is a flight simulator designed with fun in mind... it's easier to fly than Sublogic's Flight Simulator and... it is more fun to play."

John Anderson

COMPUTE!'s GAZETTE – SOLO FLIGHT

"... The word that best describes Solo Flight is Fun... it's (for those who) don't wish to spend a lot of time reading, but would rather be flying."

David Florance

NEW & IMPROVED!

For Commodore, Atari & IBM

Introducing:

- New & Improved Cockpit
- New Navigation Maps
- Software Speech (C-64 only)

Solo Flight is available for Commodore 64/128, Apple II Family, IBM PC/Jr., and Atari XL/XE and Macintosh computers. Suggested retail is only $34.95. For Macintosh only — $39.95. Call or write for more information or MC/VISA orders.

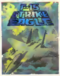
MICRO PROSE
SIMULATION • SOFTWARE

120 Lakefront Drive   Hunt Valley, Maryland 21030   301-667-1151

COMMODORE 64, ATARI, APPLE and IBM are registered trademarks of Commodore Business Inc., Atari Inc., Apple Computer Inc., and International Business Machines Inc., respectively.

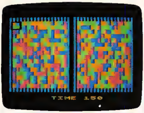
Experience the reality of these other great simulations from Micro Prose:

The Thrilling Submarine Simulation



Challenging and Exciting Air Combat from the Modern Electronic Cockpit!

You are in Command – D-Day to the Battle of the Bulge!



Play "Atari Puzzler" with a joystick.

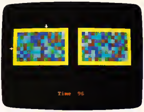
```
        {22 SPACES}":POKE DL+3
        ,68:POSITION 3,23:PRI
        NT "{4 SPACES}TIME
        {7 SPACES}";
ED 240  A=1:PB=1:OA=1:TIME=1Ø
        ØØ
FN 250  IF PEEK(7Ø4)=15 THEN
        Q2=(OA-1)*4+1:POKE 7Ø
        4,47
HK 260  GOSUB 59Ø:POKE 53248,
        Ø:FOR X=OY TO OY+7:PO
        KE X,Ø:NEXT X
KO 270  OY=Y1*8+16+YBAS*4+SPR
        :FOR X=1 TO 8:POKE OY
        +X-1,SP(X):NEXT X:POK
        E 53248,46+XBAS*4+X1*
        8
HO 280  OA=A:MAX=ROW*COL:J=ST
        ICK(Ø)-6:ON J GOTO 29
        Ø,34Ø,34Ø,34Ø,3ØØ,34Ø
        ,31Ø,32Ø:GOTO 34Ø
LP 290  A=A+1:GOTO 33Ø
LJ 300  A=A-1:GOTO 33Ø
GF 310  A=A+COL:GOTO 33Ø
FP 320  A=A-COL
BG 330  IF A>MAX OR A<1 THEN
        A=OA
NJ 340  IF STRIG(Ø)=PB THEN 3
        8Ø
JL 350  PB=STRIG(Ø):IF STRIG(
        Ø)=Ø AND PEEK(7Ø4)=Ø
        THEN POKE 7Ø4,15:GOTO
         38Ø
PC 360  IF STRIG(Ø)=Ø AND PEE
        K(7Ø4)=15 THEN Q=(A-1
        )*4+1:R=1:GOSUB 54Ø:T
        $=A$:GOSUB 57Ø:POKE 7
        Ø4,Ø:GOTO 38Ø
NB 370  IF STRIG(Ø)=Ø AND PEE
        K(7Ø4)=47 THEN GOSUB
        6ØØ:POKE 7Ø4,Ø
IB 380  POSITION 12,23:PRINT
        INT(TIME);" ";:TIME=T
        IME-Ø.1
JK 390  IF TIME<Ø THEN POSITI
        ON 3,23:PRINT "
        {3 SPACES}TIME'S UP
        ";:Z1=2Ø:Z2=7Ø:GOTO 4
        3Ø
DB 400  IF A$(1,ROW*COL*4)=B$
        (1,ROW*COL*4) THEN PO
        SITION 4,23:PRINT "YO
        U SOLVED IT";:Z1=7Ø:Z
        2=2Ø:GOTO 43Ø
GD 410  IF A<>OA THEN GOTO 25
        Ø
GJ 420  GOTO 28Ø
PI 430  GOSUB 46Ø:POSITION 2,
        23:PRINT "PRESS FIRE B
        UTTON";
PI 440  IF STRIG(Ø) THEN 44Ø
AE 450  POKE 53248,Ø:GOTO 7Ø
FC 460  FOR A=Z1 TO Z2 STEP S
        GN(Z2-Z1):SOUND Ø,A,1
        Ø,15:FOR T=A-1 TO A+1
        :SOUND 1,T,1Ø,15:NEXT
         T:POKE 712,A
AL 470  NEXT A:POKE 712,Ø:FOR
         A=15 TO Ø STEP -1:SO
        UND Ø,Z2,1Ø,A:SOUND 1
        ,Z2,1Ø,A:NEXT A:RETUR
        N
MM 480  GET #1,A:IF A<LN+48 O
        R A>HN+48 THEN 48Ø
AP 490  A=A-48:RETURN
KL 500  DATA 255,255,255,255,
        255,255,255,255,17Ø,1
        7Ø,17Ø,17Ø,17Ø,17Ø,17
        Ø,17Ø,85,85,85,85,85,
        85,85,85
JK 510  DATA 22Ø,22Ø,22Ø,22Ø,
        22Ø,22Ø,22Ø,22Ø,252,1
        32,132,132,132,132,13
        2,252
BE 520  FOR I=1 TO (COL+1)*2:
        POKE 85,9-COL+I:PRINT
         "$";:POKE 85,28-COL+
        I:PRINT "$";:NEXT I:R
        ETURN
BN 530  T$=A$(Q,Q+3):A$(Q,Q+3
        )=A$(Q2,Q2+3):A$(Q2,Q
        2+3)=T$:RETURN
PG 540  IF R=Ø THEN RETURN
IE 550  T$=A$(Q,Q+3):FOR X=1
        TO R:T2$=T$(1,1):T$(1
        ,1)=T$(3,3):T$(3,3)=T
        $(4,4):T$(4,4)=T$(2,2
        ):T$(2,2)=T2$
BM 560  NEXT X:A$(Q,Q+3)=T$:R
        ETURN
IC 570  GOSUB 59Ø:POSITION XB
        AS+X1*2,YBAS+Y1*2:PRI
        NT T$(Q,Q+1):POSITION
         XBAS+X1*2,YBAS+Y1*2+
        1
JG 580  PRINT T$(Q+2,Q+3):RET
        URN
KL 590  Z=A-1:Y1=INT(Z/COL):X
        1=Z-Y1*COL:RETURN
OQ 600  Q=(A-1)*4+1:GOSUB 53Ø
        :T$=A$:GOSUB 57Ø:ZZ=A
        :AA=Q:Q=Q2:A=(Q2-1)/4
        +1:T$=A$:GOSUB 57Ø:A=
        ZZ:Q=AA:RETURN
```

Program 5: Puzzler For IBM PC/PCjr

Version by Kevin Mykytyn, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!

```
HP 10   DEF SEG=Ø:POKE 1Ø47,64:WID
        TH 4Ø:KEY OFF:SCREEN Ø,Ø:C
        LS:X$=CHR$(219):CO(Ø)=15:C
        O(1)=14:CO(2)=12
KF 20   COLOR 12:LOCATE 5,18,Ø:PRI
        NT "Puzzler":COLOR 9:LOCAT
        E 1Ø,14:PRINT "Enter grid
        size":LN=3:HN=7
OM 30   LOCATE 14,14:PRINT "Column
        s? (3-7)":GOSUB 37Ø:COL=A
CF 40   COLOR 1Ø:LOCATE 14,14:PRIN
        T "Rows?   ":GOSUB 37Ø:ROW
        =A
EO 50   COLOR 14:LOCATE 1Ø,14:PRIN
        T "How many colors?":LN=2:
        HN=7
PM 60   LOCATE 14,14:PRINT "     (
        2-7)      ":GOSUB 37Ø:COLR=
        A
OF 70   CLS:LOCATE 1,12:PRINT "Con
        structing puzzle":LOCATE 2
```



"IBM PC/PCjr Puzzler."

```
        5,16:COLOR 11:PRINT "Pleas
        e wait";:COLOR 14
OL 80   LOCATE 12-ROW,1:GOSUB 39Ø
CO 90   FOR I=1 TO ROW*2:PRINT:PRI
        NT TAB(1Ø-COL)X$;:PRINT TA
        B(11+COL)X$;:PRINT TAB(29-
        COL)X$;:PRINT TAB(3Ø+COL)X
        $;:NEXT:PRINT:GOSUB 39Ø
MH 100  A$="":FOR A=1 TO ROW*COL*
        4:A$=A$+CHR$(INT(RND(1)*C
        OLR)+1):NEXT A:B$=A$
KN 110  FOR A=1 TO ROW*COL:Q=(A-1
        )*4+1:Q2=INT(RND(1)*ROW*C
        OL)*4+1:GOSUB 4ØØ:NEXT A
OH 120  FOR A=1 TO ROW*COL:R=RND(
        1)*4:Q=(A-1)*4+1:GOSUB 41
        Ø:NEXT:IF A$=B$ THEN 11Ø
QN 130  FOR A=1 TO ROW*COL:Q=(A-1
        )*4+1:T$=B$:XBAS=3Ø-COL:Y
        BAS=13-ROW:GOSUB 44Ø:XBAS
        =11-COL:T$=A$:GOSUB 44Ø:N
        EXT
EP 140  LOCATE 1,12:PRINT STRING$
        (2Ø,32):LOCATE 25,13:COLO
        R 12:PRINT "   Time
          ";
PB 150  A=1:PB=1:OA=1:FL=Ø:TIME=R
        OW*COL*2.5+3Ø
DO 160  IF FL=1 THEN Q2=(OA-1)*4+
        1:FL=2
KA 170  COLOR CO(FL):ZZ=A:A=OA:GO
        SUB 46Ø:A=ZZ:LOCATE YBAS+
        Y1*2,XBAS-2:PRINT " ";:LO
        CATE YBAS-2,XBAS+X1*2:PRI
        NT " ";
CJ 180  GOSUB 46Ø:LOCATE YBAS+Y1*
        2,XBAS-2:PRINT CHR$(26);:
        LOCATE YBAS-2,XBAS+X1*2:P
        RINT CHR$(25);
BA 190  OA=A:MAX=ROW*COL:K$=INKEY
        $:K$=RIGHT$(K$,1):J=ASC(K
        $+CHR$(Ø))-71:ON ABS(J) G
        OTO 2ØØ,25Ø,25Ø,21Ø,25Ø,2
        2Ø,25Ø,25Ø,23Ø:GOTO 25Ø
QK 200  A=A-COL:GOTO 24Ø
LO 210  A=A-1:GOTO 24Ø
KA 220  A=A+1:GOTO 24Ø
OP 230  A=A+COL
PL 240  IF A>MAX OR A<1 THEN A=OA
EG 250  IF J=PB THEN 29Ø
OO 260  PB=J:IF J=-58 AND FL=Ø TH
        EN FL=1:GOTO 29Ø
OM 270  IF J=-58 AND FL=1 THEN Q=
        (A-1)*4+1:R=1:GOSUB 41Ø:T
        $=A$:GOSUB 44Ø:FL=Ø:GOTO
        29Ø
EP 280  IF J=-58 AND FL=2 THEN GO
        SUB 47Ø:FL=Ø
KB 290  LOCATE 25,21:COLOR 12:PRI
        NT INT(TIME)" ";:TIME=TIM
        E-.Ø25
KA 300  IF TIME<Ø THEN LOCATE 25,
        13:PRINT "    Time's up
          ";:Z1=5ØØ:Z2=1ØØ:GOTO 3
        3Ø
```

NEW FOR YOUR apple personal computer

MacOffice: Using the Macintosh for Everything

Sharon Aker, Paul Freiberger, Arlan Levitan, Selby Bateman, Vahé Guzelimian, Dan McNeill, Charles Brannon

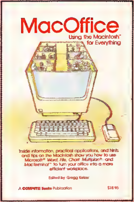
This newest release from COMPUTE! Books shows you how to get the most out of your Macintosh and software. Practical demonstrations teach you how to fully utilize several commercially available programs, such as Microsoft's *Word, Chart, File,* and *Multiplan,* Apple's *MacTerminal,* and others. You'll see how to:

- Create form letters with *Word,*
- Use *Multiplan* for your accounting,
- Share information between offices with *MacTerminal.*

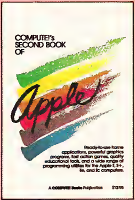
Here are the ins and outs, hints and tips on how to use each program most efficiently, and previously undocumented techniques.

Extensively illustrated with actual Macintosh-created documents and graphics, *MacOffice* shows you how you can integrate programs, let them share information, making it even easier to manage your business with a computer. You'll also learn about networks—where computers, printers, and other devices are connected to create an office-wide working environment—with details on how to create a network, and what to look for in network software.

$18.95 ISBN 0-87455-006-8



Look for these new titles at your local book or computer stores or order directly from COMPUTE! Books.



COMPUTE!'s Second Book of Apple

Edited

With over 30 programs, this collection for Apple II+, IIe, and IIc owners contains arcade-style games, games of logic and education, applications, graphics generators, and programming utilities. Compiled from the most recent issues of *COMPUTE!* magazine and *COMPUTE's Apple Applications* issue, these programs include dazzling graphics, high-speed game play, and valuable programming techniques. All the programs are ready to type in.

Applications calculate your financial situation and turn your computer into a remote terminal for accessing sophisticated databases. Action games put you in a jet fighter and in a bowling alley. Educational games range from a five-level chess game to a program which lets you write quizzes on any subject. Graphics and programming utilities help you design custom graphics or put a RAM disk in your IIc.

$12.95 ISBN 0-87455-008-4

There is also a disk available which includes the programs in the book. You can order this disk directly from COMPUTE! for only $12.95 plus $2.00 shipping charges.

COMPUTE! Books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St Anne's Road, Eastbourne, East Sussex BN 21 3UN England.

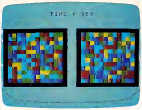
To order, call toll-free 800-334-0868 (in NC 919-275-9809) or mail your payment (including $2.00 shipping charges) to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403.

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies

```
OC 310 IF A$=8$ THEN LOCATE 25,1
       5:PRINT "You solved it";:
       Z1=100:Z2=500:GDTD 330
NF 320 IF A<>DA THEN 160 ELSE IF
        J=-58 THEN 170 ELSE 190
PB 330 GDSUB 360:LDCATE 25,13:PR
       INT " Press any key  ";
EM 340 A$=INKEY$:IF A$="" THEN 3
       40
GH 350 RUN
PD 360 FDR A=Z1 TD Z2 STEP 20*SG
       N(Z2-Z1):SDUND A,2:CDLDR
       Ø,Ø,RND(1)*6+1:FDR TD=1 T
       D 99:NEXT:NEXT:CDLOR 15,Ø
       ,Ø:RETURN
LE 370 Z=RND(1):K$=INKEY$:A=VAL(
       K$):IF A<LN OR A>HN THEN
       37Ø
NN 380 RETURN
QI 390 FDR I=1 TD (CDL+1)*2:LDCA
       TE ,9-COL+I:PRINT X$;:LOC
       ATE ,28-CDL+I:PRINT X$;:N
       EXT I:RETURN
BK 400 T$=MID$(A$,Q,4):MID$(A$,Q
       ,4)=MID$(A$,Q2,4):MIO$(A$
       ,Q2,4)=T$:RETURN
BG 410 IF R=Ø THEN RETURN
PK 420 T$=MID$(A$,Q,4):FDR X=1 T
       O R:T2$=MIO$(T$,1,1):MID$
       (T$,1,1)=MID$(T$,3,1):MIO
       $(T$,3,1)=MID$(T$,4,1):MI
       D$(T$,4,1)=MID$(T$,2,1):M
       ID$(T$,2,1)=T2$
OF 430 NEXT:MID$(A$,Q,4)=T$:RETU
       RN
GC 440 GOSUB 460:LOCATE Y8AS+Y1*
       2,XBAS+X1*2:CDLOR ASC(MIO
       $(T$,Q,1)):PRINT X$;:CDLO
       R ASC(MIO$(T$,Q+1,1)):PRI
       NT X$;:LOCATE Y8AS+Y1*2+1
       ,XBAS+X1*2
DH 450 COLOR ASC(MID$(T$,Q+2,1))
       :PRINT X$;:COLOR ASC(MIO$
       (T$,Q+3,1)):PRINT X$;:RET
       URN
EB 460 Z=A-1:Y1=INT(Z/COL):X1=Z-
       Y1*COL:RETURN
CP 470 Q=(A-1)*4+1:GOSU8 400:T$=
       A$:GOSUB 440:ZZ=A:AA=Q:Q=
       Q2:A=(Q2-1)/4+1:T$=A$:GOS
       UB 440:A=ZZ:Q=AA:RETURN
```

Program 6: TI-99/4A Puzzler

Version by Patrick Parrish, Programming Supervisor



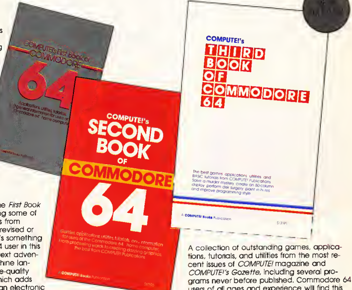
"Puzzler" for the TI-99/4A can be played with a joystick or the keyboard.

```
100 RANOOMIZE :: CALL CLE
    AR :: GOSUB 38Ø :: CA
    LL MAGNIFY(3)
110 GOSUB 37Ø :: CALL SCR
    EEN(2):: OISPLAY AT(7
    ,11):"PUZZLER !" :: O
    ISPLAY AT(12,4):"INPU
    T GRID SIZE (3-6)
120 OISPLAY AT(14,11):"RO
    WS ?" :: ACCEPT AT(14
    ,1B):R :: IF R<3 OR R
    >6 THEN 12Ø
130 OISPLAY AT(16,1Ø):"CO
    LUMNS ?" :: ACCEPT AT
    (16,2Ø):C :: IF C<3 O
    R C>6 THEN 13Ø
140 OISPLAY AT(18,3):"HOW
     MANY COLORS (2-6)?"
    :: ACCEPT AT(18,26):C
    OLS :: IF COLS<2 OR C
    OLS>6 THEN 14Ø
150 CALL CLEAR :: FDR I=1
     TO B :: CALL COLOR(I
    ,2,1):: NEXT I :: CAL
    L SCREEN(15):: U=C*2+
    3 :: U=INT((19-U)/2):
    : D=R*C*1Ø
160 TE=12-R :: OISPLAY AT
    (2,5):"CDNSTRUCTING P
    UZZLE" :: DISPLAY AT(
    23,9):"PLEASE WAIT"
170 MT=C*2+2 :: CALL HCHA
    R(TE,U,35,MT):: CALL
    HCHAR(TE,U+16,35,MT):
    : A=R*2
180 CALL VCHAR(TE+1,U,35,
    A):: CALL VCHAR(TE+1,
    U+C*2+1,35,A):: CALL
    VCHAR(TE+1,U+16,35,A)
    :: CALL VCHAR(TE+1,U+
    C*2+17,35,A)
190 CALL HCHAR(A+TE+1,U,3
    5,MT):: CALL HCHAR(A+
    TE+1,U+16,35,MT):: Y=
    TE+1 :: X=U
200 A$="" :: FOR I=1 TO R
    *C*4 :: RANOOMIZE ::
    A$=A$&CHR$(INT(RNO*CO
    LS)*8+96):: NEXT I ::
     B$=A$ :: FOR I=1 TO
    R*C
210 R1=INT(R*C*RNO)*4+1 :
    : R2=INT(R*C*RNO)*4+1
     :: IF R1=R2 THEN 21Ø
220 TEM$=SEG$(A$,R1,4)::
    TEM2$=SEG$(A$,R2,4)::
     GOSUB 49Ø :: NEXT I
    :: FOR T=1 TO R*C*4-3
     STEP 4
230 TEM$=SEG$(A$,T,4):: R
    1=INT(RNO*4):: FOR J=
    1 TO R1 :: GOSUB 52Ø
    :: NEXT J :: 8OSU8 53
    Ø :: NEXT T :: IF A$=
    8$ THEN 2ØØ
240 FOR I=Ø TO R-1 :: FOR
     J=Ø TO C-1 :: GOSUB
    42Ø
250 OISPLAY AT(Y+2*I,X+2*
    J+15):SEG$(8$,J*4+1+I
    NT((2*I+1)/2)*C*4,2);
    :: OISPLAY AT(Y+2*I+1
    ,X+2*J+15):SEG$(B$,J*
    4+3+INT((2*I+1)/2)*C*
    4,2);
260 NEXT J :: NEXT I :: C
    ALL HCHAR(2,7,32,19):
    : CALL HCHAR(23,11,32
    ,11):: SC=2 :: LY=TE*
    B+1 :: LX=U*B+1 :: SY
    =LY :: SX=LX :: Q=1 :
    : F,I,J=Ø
270 OISPLAY AT(2,1Ø):"TIM
    E :";O
280 CALL SPRITE(#1,1ØØ,CS
    (F),SY,SX):: O=D-.25
    :: OISPLAY AT(2,16):I
    NT(O):: IF INT(D)=Ø T
    HEN GDTD 34Ø
290 CALL KEY(Ø,K,ST):: CA
    LL KEY(1,KK,ST):: IF
    ST=Ø THEN CALL JOYST(
    1,H,V):: H=SGN(H):: V
    =SGN(-V)ELSE H=(K=83)
    -(K=68):: V=(K=69)-(K
    =88)
300 J=J+H :: I=I+V :: J=J
    +(J>C-1)*C-(J<Ø)*C ::
     I=I+(I>R-1)*R-(I<Ø)*
    R :: SX=LX+J*16 :: SY
    =LY+I*16 :: IF KK=18
    DR K=32 THEN GDSUB44Ø
310 IF (DX<>SX OR OY<>SY)
    ANO F=1 THEN F=2 :: G
    OSU8 47Ø
320 IF A$<>B$ THEN 28Ø
330 FDR I=1 TD 3Ø STEP 3
    :: CALL SOUND(75,22Ø+
    2Ø*I,4):: CALL SCREEN
    (INT(I/2)+1):: NEXT I
     :: REM WIN GAME
340 FDR I=3Ø TO 1 STEP -3
     :: CALL SDUND(75,22Ø
    +2Ø*I,4):: CALL SCREE
    N(INT(I/2)+1):: NEXT
    I :: CALL SCREEN(15)
350 DISPLAY AT(23,6):"PLA
    Y AGAIN (Y/N)?" :: AC
    CEPT AT(23,24)BEEP VA
    LIDATE("YNyn"):A$ ::
    IF A$="N" OR A$="n" T
    HEN STDP
360 CALL OELSPRITE(#1)::
    GDTO 11Ø
370 CALL CLEAR :: FOR I=1
     TO 8 :: CALL COLOR(I
    ,16,1):: NEXT I :: RE
    TURN
380 CALL CHAR(1ØØ,"FFBØBØ
    8Ø8Ø8ØBØBØ8ØBØBØ8ØBØB
    Ø8ØFFFFØ1Ø1Ø1Ø1Ø1Ø1Ø1
    Ø1Ø1Ø1Ø1Ø1Ø1FF")
390 FDR I=96 TO 136 STEP
    8 :: CALL CHAR(I,"FFF
    FFFFFFFFFFFFF"):: NEX
    T I
400 FOR I=9 TO 14 :: READ
     A :: CALL COLOR(I,A,
    1):: NEXT I :: CALL C
    HAR(35,RPT$("F",16)):
    : FOR F=Ø TO 2 :: REA
    O CS(F):: NEXT F :: R
    ETURN
410 DATA 3,5,7,B,11,14,2,
    16,1Ø
420 DISPLAY AT(Y+2*I,X+2*
    J-1):SEG$(A$,J*4+1+IN
    T((I*2+1)/2)*C*4,2);:
    : OISPLAY AT(Y+2*I+1,
    X+2*J-1):SEG$(A$,J*4+
    3+INT((2*I+1)/2)*C*4,
    2);
430 RETURN
440 IF F=Ø THEN OX=SX ::
    OY=SY :: GOSUB 51Ø ::
     R1=T :: F=1 :: GDSUB
     47Ø :: OJ=J :: OI=I
    :: RETURN
450 IF F=1 THEN GOSUB 51Ø
     :: TEM$=SEG$(A$,T,4)
    :: GOSUB 52Ø :: GOSUB
     53Ø :: GOSU8 42Ø ::
    F=Ø :: GOSU8 47Ø :: R
    ETURN
460 GDSU8 51Ø :: R2=T ::
    GDSUB 48Ø :: GOSUB 42
    Ø :: TJ=J :: TI=I ::
    I=OI :: J=OJ :: GOSUB
```

A *SPECIAL OFFER* from COMPUTE! Books

for Owners and Users of Commodore Computers

Buy any 2 of these books and receive a ***15% Discount***. You pay $22.00 and save $3.90!
Buy all 3 books and get a ***25% Discount***. You pay only $29.00, a savings of $9.85!



An excellent resource for users of the 64, with something for everyone: BASIC programming techniques, a memory map, a machine language monitor, and information about writing games and using peripherals. This 264-page, spiral-bound book includes many ready-to-type-in programs and games. $12.95

Continues in the tradition of the *First Book of Commodore 64* in presenting some of the best programs and articles from COMPUTE! Publications, many revised or never before published. There's something for almost any Commodore 64 user in this 289-page book: arcade and text adventure games in BASIC and machine language, a commercial software-quality word processor, a program which adds 41 new commands to BASIC, an electronic spreadsheet, tutorials about programming sound and graphics, and utilities for saving, copying, and retrieving files. $12.95

A collection of outstanding games, applications, tutorials, and utilities from the most recent issues of *COMPUTE!* magazine and *COMPUTE!'s Gazette*, including several programs never before published. Commodore 64 users of all ages and experience will find this book informative, entertaining, and educational. Create an 80-column display, play educational and arcade-quality games, compose music, move sprites easily and quickly, and see how to program more efficiently and effectively. $12.95

Buy Now—This Offer Expires December 15, 1985.

Yes! I want to save money while I enjoy COMPUTE! Books.
_______ COMPUTE!'s First Book of Commodore 64, $12.95
_______ COMPUTE!'s Second Book of Commodore 64, $12.95
_______ COMPUTE!'s Third Book of Commodore 64, $12.95

All Orders Must Be Prepaid
☐ Payment Enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ VISA ☐ American Express
Acct. No. ________________ Exp. Date ________
Signature ________________
Name ________________
Address ________________
City ________________
State ________________ Zip ________

To Order Call Toll Free **800-334-0868**
(in NC call **919-275-9809**)
or mail this coupon with your payment to:
COMPUTE! Books, P.O. Box 5406,
Greensboro, NC 27403

Please send me:
☐ 1 Book for $12.95
☐ 2 Books for $22.00
☐ 3 Books for $29.00

NC residents add 4.5% sales tax ________
Add $2.00 per book for shipping ________
Total Paid ________

Please allow 4-6 weeks for delivery. 75D1121

```
     420 :: F=Ø :: GOSUB
    470 :: J=TJ :: I=TI :
    : RETURN
470 CALL COLOR(#1,CS(F)):
    : RETURN
480 IF R1=R2 THEN RETURN
    :: REM TRANSPOSE
490 IF R2>R1 THEN A=R1 ::
     B=R2 ELSE A=R2 :: B=
    R1
500 A$=SEG$(A$,1,A-1)&SEG
    $(A$,B,4)&SEG$(A$,A+4
    ,B-A-4)&SEG$(A$,A,4)&
    SEG$(A$,B+4,LEN(A$)-B
    +5):: RETURN
510 T=J*4+1+INT((I*2+1)/2
    )*C*4 :: RETURN :: RE
    M CALC STRING POINTER
520 TEM$=SEG$(TEM$,3,1)&S
    EG$(TEM$,1,1)&SEG$(TE
    M$,4,1)&SEG$(TEM$,2,1
    ):: RETURN :: REM ROT
    ATE
530 A$=SEG$(A$,1,T-1)&TEM
    $&SEG$(A$,T+4,LEN(A$)
    -T-3):: RETURN :: REM
     SUBSTITUTE ROTATED S
    UBSTRING
```

"Puzzler" runs on any Apple II-series computer.

Program 7: Apple Puzzler

Version by Kevin Martin, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!

```
29 100 A$ = "": IF PEEK (24576)
        = 162 THEN 140
55 110 FOR I = 24576 TO 24872
02 120 READ A: POKE I,A
FE 130 NEXT
C5 140 HIMEM: 24576
50 150 GOSUB 550
50 160 IF T = Ø THEN VTAB 21: PR
        INT TAB( 14)"OUT OF TIME"
        : GOTO 380
DB 170 HTAB 17: VTAB 23: PRINT T
        ;" "
47 180 T = T - 1
2A 190 IF PEEK ( - 16384) < 128
        THEN 160
90 200 GET C$: IF (C$ < "I" OR C
        $ > "L") AND C$ < > " " T
        HEN 160
01 210 R = R - (C$ = "I") + (C$
        = "K")
59 220 C = C - (C$ = "J") + (C$
        = "L")
7E 230 IF R < Ø THEN R = Ø
9E 240 IF R > = R3 THEN R = R3 -
         1
AF 250 IF C < Ø THEN C = Ø
DE 260 IF C > = C3 THEN C = C3 -
         1
61 270 POKE 773,X1 + C * 2 - 1:
        POKE 772,Y1 + R * 2 - 1:
        CALL 24671
C0 280 IF C$ < > " " THEN 160
46 290 IF F = Ø THEN 440
0A 300 F = Ø: IF RR = R AND CC =
         C THEN GOSUB 510: GOTO 3
        2Ø
4B 310 GOSUB 460
02 320 CALL 24691
C4 330 POKE 768,X1: POKE 769,Y1:
         CALL 24576
80 340 POKE 773,X1 + C * 2 - 1:
        POKE 772,Y1 + R * 2 - 1:
        POKE 774,255: CALL 24753
C3 350 IF A$ < > B$ THEN 160
DA 360 CALL 24691
74 370 HOME : PRINT TAB( 16);"CO
        RRECT!"
59 380 HTAB 13: VTAB 22: PRINT "
        PRESS ANY KEY."
1C 390 HTAB 17: VTAB 23: PRINT T
0F 400 POKE - 16368,Ø
0F 410 IF PEEK ( - 16384) < 128
        THEN 410
D4 420 GET A$
AC 430 RUN
3D 440 F = 1:RR = R:CC = C: POKE
         773,X1 + C * 2 - 1: POKE
         772,Y1 + R * 2 - 1: POKE
         774,119: CALL 24671
9D 450 GOTO 160
8B 460 AA = SS + 2 * NC * RR + 2
         * CC:A = SS + 2 * NC * R
         + C * 2
04 470 D = PEEK (A): POKE A, PEE
        K (AA): POKE AA,D
C0 480 D = PEEK (A + 1): POKE A
        + 1, PEEK (AA + 1): POKE
        AA + 1,D
98 490 D = PEEK (A + NC): POKE A
         + NC, PEEK (AA + NC): PO
        KE AA + NC,D
A4 500 D = PEEK (A + NC + 1): PO
        KE A + NC + 1, PEEK (AA +
         NC + 1): POKE AA + NC +
        1,D: RETURN
4E 510 A = SS + 2 * NC * R + C *
         2
40 520 D = PEEK (A): POKE A, PEE
        K (A + NC)
4E 530 POKE A + NC, PEEK (A + NC
         + 1)
6F 540 POKE A + NC + 1, PEEK (A
        + 1): POKE A + 1,D: RETUR
        N
62 550 TEXT : HOME
00 560 PRINT TAB( 16);"PUZZLER"
F2 570 INPUT "NUMBER OF ROWS (2-
        7):";R3
00 580 IF R3 < 2 OR R3 > 7 THEN
        57Ø
77 590 INPUT "NUMBER OF COLUMNS
        (2-7):";C3
6F 600 IF C3 < 2 OR C3 > 7 THEN
        59Ø
8C 610 INPUT "NUMBER OF COLORS (
        2-15):";CO
FD 620 IF CO < 2 OR CO > 15 THEN
         61Ø
0D 630 PRINT "PLEASE WAIT..."
FC 640 NR = 2 * R3:NC = 2 * C3
7E 650 FOR A = 1 TO NR * NC:B =
        INT ( RND (1) * CO + 1):A
        $ = A$ + CHR$ (B + 8 * 16
        ): NEXT :B$ = A$
A2 660 A = PEEK (105) + PEEK (10
        6) * 256
CF 670 SS = PEEK (A + 3) + PEEK
        (A + 4) * 256
54 680 X1 = 1Ø - C3:Y1 = 9 - R3:
        X2 = X1 + 2Ø
50 690 POKE 24600, PEEK (A + 3):
         POKE 24601, PEEK (A + 4)
96 700 POKE 768,X2: POKE 769,Y1:
         POKE 770,NC: POKE 771,NR
         + Y1
49 710 GR
F4 720 CALL 24576
ED 730 FOR R = Ø TO R3 - 1: FOR
        C = Ø TO C3 - 1:B = INT (
         RND (1) * 4)
4B 740 IF B THEN GOSUB 510:B = B
         - 1: GOTO 74Ø
CA 750 NEXT : NEXT
42 760 FOR R = Ø TO R3 - 1: FOR
        C = Ø TO C3 - 1
04 770 RR = INT ( RND (1) * R3):
        CC = INT ( RND (1) * C3):
         GOSUB 460: NEXT : NEXT
02 780 POKE 768,X1: POKE 769,Y1:
         CALL 24576
5E 790 HOME : PRINT TAB( 16);"PU
        ZZLER"
20 800 POKE 772,Y1 - 1: POKE 773
        ,X1 - 1: POKE 774,255: CA
        LL 24753
03 810 R = Ø:C = Ø:T = NR * NC *
         75: RETURN
0E 820 DATA 162,Ø,172,1,3,185
C9 830 DATA 47,96,24,1Ø9,Ø,3
93 840 DATA 133,251,185,71,96,1Ø
        5
72 850 DATA Ø,133,252,16Ø,Ø,189
08 860 DATA 14Ø,89,145,251,232,2
        ØØ
AF 870 DATA 2Ø4,2,3,2Ø8,244,238
70 880 DATA 1,3,173,1,3,2Ø5
DD 890 DATA 3,3,2Ø8,212,96,Ø
7E 900 DATA 128,Ø,128,Ø,128,Ø
39 910 DATA 128,4Ø,168,4Ø,168,4Ø
F5 920 DATA 168,4Ø,168,8Ø,2Ø8,8Ø
05 930 DATA 2Ø8,8Ø,2Ø8,8Ø,2Ø8,4
4A 940 DATA 4,5,5,6,6,7
21 950 DATA 7,4,4,5,5,6
1E 960 DATA 6,7,7,4,4,5
3E 970 DATA 5,6,6,7,7,32
CA 980 DATA 115,96,76,177,96,24
E3 990 DATA 121,47,96,133,251,18
        5
4C 1000 DATA 71,96,1Ø5,Ø,133,252
0B 1010 DATA 96,172,7,3,173,8
FA 1020 DATA 3,32,1Ø1,96,16Ø,Ø
67 1030 DATA 162,Ø,189,9,3,145
D1 1040 DATA 251,232,2ØØ,2ØØ,2ØØ
        ,189
A5 1050 DATA 9,3,145,251,232,173
0C 1060 DATA 7,3,24,1Ø5,3,141
31 1070 DATA 7,3,168,173,8,3
72 1080 DATA 32,1Ø1,96,16Ø,Ø,189
9C 1090 DATA 9,3,145,251,232,2ØØ
CD 1100 DATA 2ØØ,2ØØ,189,9,3,145
0D 1110 DATA 251,232,96,172,4,3
CA 1120 DATA 14Ø,7,3,173,5,3
09 1130 DATA 141,8,3,32,1Ø1,96
27 1140 DATA 16Ø,Ø,162,Ø,177,251
67 1150 DATA 157,9,3,232,41,15
0D 1160 DATA 145,251,173,6,3,41
51 1170 DATA 24Ø,17,251,145,251,
        2ØØ
1F 1180 DATA 2ØØ,2ØØ,177,251,157
        ,9
FE 1190 DATA 3,232,41,15,145,251
11 1200 DATA 173,6,3,41,24Ø,17
3A 1210 DATA 251,145,251,173,4,3
F0 1220 DATA 24,1Ø5,3,141,4,3
AF 1230 DATA 168,173,5,3,32,1Ø1
2F 1240 DATA 96,16Ø,Ø,177,251,15
        7
DF 1250 DATA 9,3,232,41,24Ø,145
31 1260 DATA 251,173,6,3,41,15
0F 1270 DATA 17,251,145,251,2ØØ,
        2ØØ
EE 1280 DATA 2ØØ,177,251,157,9,3
96 1290 DATA 232,41,24Ø,145,251,
        173
C0 1300 DATA 6,3,41,15,17,251
23 1310 DATA 145,251,96
```

©

"If you know BASIC and want to learn machine language, this is the place to start .... Building on your experience as a BASIC programmer, Mansfield very gently takes you through the fundamentals of machine language."

—Whole Earth Software Catalog

COMPUTE! Books'

Best-selling Machine Language Books

"Understandable"—**The New York Times**

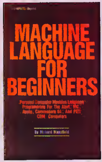
"Presents the machine language novice with a very good tutorial in simple, understandable terms."

—**Antic**

"I highly recommend *Machine Language for Beginners* as your first introduction to the world of machine language." —**Commodore Power/Play**

The LADS Disk

LADS, the assembler used in *The Second Book of Machine Language*, is available on disk for only $12.95. This is a great accompaniment to the book, saving you hours of typing time by providing the complete source and object programs for all versions of the assembler, and more. And LADS disks are specific to your Apple, Atari, or Commodore computers.



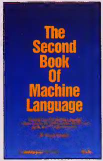
Machine Language for Beginners

Richard Mansfield

Most commercial software is written in machine language because it's far faster and more versatile than BASIC. *Machine Language for Beginners* is a step-by-step introduction. Includes a simple assembler, a disassembler, and utilities, to help beginners write programs more quickly and easily.

$14.95

ISBN 0-942386-11-6



The Second Book of Machine Language

Richard Mansfield

The follow-up to the best-selling *Machine Language for Beginners*, this book leads the programmer deeper into the most powerful and efficient programming techniques available for personal computers. Fully tutorial, with easy step-by-step explanations, the book shows how to construct significant, effective machine language programs. Included is a high-speed, professional-quality, label-based assembler. Everything that's needed for optimized programming on the Commodore 64, Apple, Atari, VIC-20, and PET/CBM computers.

$14.95

ISBN 0-942386-53-1

Machine Language for Beginners and *The Second Book of Machine Language*: everything you need to learn machine language programming on your Apple, Atari, and Commodore personal computers.

To Order: Call Toll Free **800-334-0868** (In NC **919-275-9809**) or mail this coupon with your payment to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403.

_____ *The Second Book of Machine Language*, **$14.95**
_____ *Machine Language for Beginners*, **$14.95**
_____ *LADS Disk* (Apple) **$12.95**
_____ *LADS Disk* (Atari) **$12.95**
_____ *LADS Disk* (Commodore), **$12.95**

☐ Payment Enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa ☐ American Express

Acct. No. ____________

Signature ____________

Name ____________

Address ____________

City ____________

State ______ Zip ______

_____	1 Book for	$14.95
_____	2 Books for	$25.00
_____	*LADS Disk* for	$12.95
NC residents add 4.5% sales tax		$_____
Shipping and handling ($2.00 per book $1.00 per disk)		$_____
Total Paid		$_____

All orders must be prepaid
Please allow 4–6 weeks for delivery

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies

75D1111

HOTWARE: Software Best Sellers

This Month	Last Month	Title	Publisher	Remarks	Apple	Atari	Commodore	IBM	Macintosh
Entertainment									
1.	2.	*F-15 Strike Eagle*	MicroProse	Air combat simulation	•	•	•	•	
2.	3.	*Flight Simulator II*	SubLogic	Aircraft simulation	•	•	•		
3.		*Gato*	Spectrum Holobyte	Submarine simulation	•			•	
4.	5.	*Karateka*	Brøderbund	Action karate game	•		•		
5.	4	*Flight Simulator*	Microsoft	Aircraft simulation				•	
Education									
1.	2.	*Typing Tutor III*	Simon & Schuster	Typing instruction program	•		•	•	•
2	3	*Math Blaster!*	Davidson	Introductory math program, ages 6-12	•	•	•	•	
3.	1.	*New Improved MasterType*	Scarborough	Typing instruction program	•	•	•	•	•
4.		*Charlie Brown's ABC's*	Random House	Letter and word introduction, ages 3-4	•	•		•	
5.	5.	*Music Construction Set*	Electronic Arts	Music composition program	•	•	•		
Home Management									
1.	1.	*Print Shop*	Brøderbund	Do-it-yourself print shop	•	•	•		
2.	4.	*The Newsroom*	Springboard	Do-it-yourself newspaper	•		•	•	
3.	2.	*Print Shop Graphics Library*	Brøderbund	100 additional graphics	•	•	•		
4.		*Print Master*	Unison World	At-home print shop				•	
5.		*Bank Street Writer*	Brøderbund	Word processor	•	•	•	•	

Skyscape

Robert M. Simons

This unique program, written by a planetarium director, presents the sky as it can be viewed at any date and time from the year 1977 onward—including zodiac constellations and all the visible planets. It also calculates planet tables, positions of the sun, and phases of the moon for any date and time from 1977 into the future. As an extra (and timely) bonus, it can even display Halley's Comet, due to become visible in late 1985 and early 1986. "Skyscape" is both educational and entertaining. The original version is for the Commodore 64, and we've written additional versions for Apple II-series computers with DOS 3.3 or ProDOS; the TI-99/4A with Extended BASIC; the IBM PC with color/graphics adapter; the PCjr with Cartridge BASIC; and Atari 400/800, XL, and XE computers with at least 24K RAM for tape or 32K for disk.

For thousands of years the sun, moon, and planets in our solar system have excited human imagination. In ancient times they were regarded as gods whose distant motions influenced the course of earthly events. Though we now understand more about the true nature of celestial objects, many facts remain unknown, and a brilliant nighttime sky still presents an inspiring spectacle.

Whether you're seriously interested in the sky or just casually curious, "Skyscape" is a convenient tool for extending your knowledge. It opens a movable window on the heavens, displaying the position of our sun, moon, and neighboring planets from almost any location on Earth, at any point in time from 1977 into the distant future. Since it performs all the necessary calculations, you can enjoy and learn from this program even if you're not an expert in astronomy. In addition to providing data about the position of celestial objects, it draws a sky map on the screen, showing each object as it would appear to you at the chosen location and time.

To get started, type in the appropriate version of Skyscape for your computer and save a copy before running it.

Past, Present, Or Future

Skyscape begins by asking you to answer several questions. Enter the year, choosing any year from 1977 forward. In some ways this is the most important input of all, since objects in our solar system move significantly from one year to the next. After you choose the year, Skyscape allows you to enter the month and day.

Next you must enter the latitude (north/south position on Earth) from which you wish to view the sky. Latitude 0 places you, the observer, at the equator. Latitudes 1–90 place you in the northern hemisphere (north of the equator). To choose a southern latitude (south of the equator), enter a negative number from −1 to −90. Skyscape generally represents southerly locations with negative values.

Whenever Skyscape asks for information, it checks your entry to make sure it's in the acceptable range. If you enter an illegal value, the program displays an error message and gives you another chance.

The Sun And Moon

Though very different in size and composition, the sun and moon are alike in being the largest celestial objects visible from Earth. After you enter the date and latitude, Skyscape displays a table of data for the sun and moon. In addition to the date, day of the year, and latitude north or south, you'll see the following information:

• Sun's geocentric angle. This figure represents the sun's position as a number of degrees relative to the vernal equinox. The vernal equinox is where the sun is located when spring begins in the northern hemisphere (the same time that autumn begins in the southern hemisphere).

• Sun's declination. The number of degrees north or south of the equator. Negative values indicate a southerly location.

• Sun's altitude at noon. The location of the sun in degrees from the northern or southern horizon at noon.

• Sun's right ascension. Just as longitude and latitude indicate locations on the Earth, *right ascension* and *declination* are used to pinpoint locations in the sky. For this purpose the sky is visualized as a gigantic sphere surrounding the Earth. Declination locates a point vertically in the celestial sphere and right ascension locates it horizontally. Right ascension values are given in *hours* and *minutes* in the range 0:00–23:59. Right ascension 0:00 is exactly at the vernal equinox. Larger right ascension values lie to the east of smaller ones.

• Right ascension at 9 p.m. The right ascension which would be on the meridian at 9 p.m. This coordinate system would be found on star charts. By comparing this number with those charts, you can tell what stars and constellations would be visible at that time.

• Moon's age. The number of days since the last new moon.

• Moon's elongation. The location of the moon in degrees east or west of the sun.

• Moon's phase. The phase of the moon on this particular day.

The Planet Table

After viewing the sun and moon display, press P to continue to the next display screen, which contains the planet table. (Press D if you wish to enter a new date.) The planet table shows vital information about the visible planets (through Uranus, which is at the limit of our visibility). The table shows the position of each planet in right ascension and degrees east or west of the sun. It also shows the distance of each planet from Earth in millions of miles.

If you'd rather see the distance in kilometers, modify the program to change the value of ES=93 to ES=149.6 (the program line which defines the value of ES varies with the version of Skyscape: Commodore 64—line 220; Atari—line 190; IBM—line 130; Apple—line 80; TI—line 150).

Some planets have an asterisk to the left of the right ascension figure. This signifies that they are visible at 9 o'clock this evening. For reference, the planet table also includes the sun's present right ascension and its right ascension at 9 p.m. Press D to input a new date or S to view a graphics display of the sky at any time in the current day.

The Visible Skyscape

After selecting the sky display, you must enter the hour when you wish to view the sky. The hour value should be a whole number from 0–23 (enter 22 for 10 p.m., etc.). You'll also need to enter the minutes (0–59). Skyscape then displays the time and offers you a chance to enter different values. Press RETURN or Enter when you're satisfied with the time.

Skyscape now displays the sky as it would appear at the chosen latitude, date, and time. Since the sky looks very different from different places on Earth, the latitude affects the display considerably. If your latitude is in the range 24–90 degrees north or south, the sky shows a dashed line representing the position of the celestial equator, along with symbols representing the sun, moon, and planets visible at that time. If your latitude is in the tropical region—from 23½ degrees north to 23½ degrees south—the dashed line indicates a position directly overhead.

If you're viewing in the northern hemisphere, north is above the dashed line and south is below it. In the southern hemisphere these directions are reversed. Below the sky display is a key that interprets the symbols used to represent celestial objects. If more than one object is positioned at the same spot, the symbols are displayed above each other.

At the bottom of the sky you may see two-letter abbreviations. These represent zodiac constellations that would be visible from your chosen vantage point. Skyscape uses the abbreviations AR (Aries), PI (Pisces), AQ (Aquarius), CP (Capricorn), SA (Sagittarius), SC (Scorpio), LI (Libra), VI (Virgo), LE (Leo), CA (Cancer), GE (Gemini) and TA (Taurus). Each constellation is located above the spot where its abbreviation appears. In northern latitudes, the border of each constellation's zone begins at its abbreviation and extends left. In southern latitudes, the constellation extends right from the position of its abbreviation.

Daytime skies are shown in blue and nighttime skies in black. Skyscape does not calculate the actual rising or setting time of the sun. Average rising and setting times of 6 a.m. and 6 p.m. are used in every case. You may obtain exact rising and setting times from local newspapers. However, keep in mind that there is usually about an hour of twilight before sunrise and after sunset.

Halley's Comet

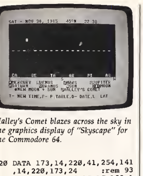
In addition to permanent objects, Skyscape's graphics display includes Halley's Comet, which should be visible during late 1985 and early 1986. If you choose a date from November 1, 1985 to May 29, 1986, Skyscape calculates the position of Halley's Comet and includes it in the graphics display (if it would be visible at the place and time you select). The comet's position is based on the best predictions available at the time of this writing (summer 1985). These positions may differ slightly from the comet's actual position when it finally makes its appearance.

While Skyscape is generally accurate, it bases most position calculations on circular orbits. This introduces a certain element of error, since no object in our solar system has a perfectly circular orbit. The position error is most pronounced for Mercury and Mars (whose orbits are quite elliptical), but does not significantly affect other objects. I've found Skyscape accurate enough for my own purposes, which include planning astronomy classes and planetarium displays.

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!

Program 1: Commodore 64 Skyscape

```
100 POKE56,56:POKE55,0:CLR:FOR
    I=828TO909:READA:X=X+A:POK
    EI,A:NEXTI:PRINTCHR$(8)
                        :rem 246
110 IFX<>9923THENPRINT"ERROR I
    N DATA STATEMENTS.":STOP
                        :rem 187
```



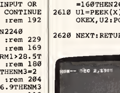
Halley's Comet blazes across the sky in the graphics display of "Skyscape" for the Commodore 64.

```
120 DATA 173,14,220,41,254,141
    ,14,220,173,24       :rem 93
130 DATA 208,41,14,10,10,133,1
    67,169,208,133       :rem 102
140 DATA 252,173,0,221,41,3,73
    ,3,10,10,10,10       :rem 69
150 DATA 10,10,5,167,133,254,1
    65,1,41,251          :rem 205
160 DATA 133,1,169,0,133,251,1
    33,253,168,162       :rem 109
170 DATA 8,177,251,145,253,200
    ,208,249,230         :rem 21
180 DATA 252,230,254,202,208,2
    42,165,1,9           :rem 168
190 DATA 4,133,1,173,14,220,9,
    1,141,14,220,96      :rem 144
200 POKE53281,1:POKE646,0:GOSU
    82210                :rem 5
210 D$="000031059090120151181
    2 12243273304334":K1=1440:DI
    MHC(22):MM$="041081040"
                         :rem 225
220 M$="28631734501104107210 21
    33164194225255":D$(1)="S":
    D$(2)="N":ES=93      :rem 28
230 A$="JANFEBMARAPRMAYJUNJULA
    UGSEPOCTNOVDEC":OO$="
    {DOWN}OUT OF RANGE!!{DOWN}
    "                    :rem 232
240 MD$="312831303130313130313
    031":D9=↑/180:READEE:READM
    9:DIMP(6,6)          :rem 66
250 DEFFNR(X)=INT(X*100+.5)/10
    0                    :rem 207
260 DEFFNS(X)=INT(X*10+.5)/10
                         :rem 113
270 FORY=1TO2:FORX=1TO6:READP(
    X,Y):NEXT:NEXT:Y=0:rem 162
280 FORX=1TO6:READP$(X),P(X,3)
    :NEXT:FORX=1TO8:READA:POKE
    14335+X,A:NEXT       :rem 187
290 FORX=15024TO15079:READA:PO
    KEX,A:NEXT:FORX=1TO7:PP(X)
    =X+85:NEXT           :rem 228
300 J$="SATSUNMONTUEWEDTHUFRI"
    :FORX=1TO12:READF$:rem 151
310 CC$=CC$+"{5 SPACES}"+F$:NE
    XT:CC$=CC$+CC$:F$=RIGHT$(C
    C$,9):CC$=F$+CC$   :rem 133
320 FORX=1TO8:READPH$(X):NEXT
                         :rem 81
330 FORX=1TO22:READHC(X):NEXT:
    POKE53281,7:GOTO920:rem 42
340 CC=MT-720:IFCC<0THENCC=CC+
    K1                   :rem 155
350 CC=CC/120:CD=CC-INT(CC):CC
    =INT(CC):CD=INT(CD*7+.2):C
    C=81-(CC*7+CD)       :rem 255
360 GOSUB2000:PRINTCHR$(18);CD
    $;CHR$(146);:IFLL<0THENGOS
    UB2590               :rem 242
370 FORX=55976TO56015:POKEX,2:
    GOTO2620             :rem 38
380 PRINT"{CLR}{DOWN}"TAB(10)"
    ** DAYS SKY **":GOSUB1770:
    PRINT                :rem 253
390 PRINT:PRINT"INPUT THE TIME
    :":PRINT"E15 Y3":T1=0:T2=0
                         :rem 43
400 PRINT:INPUT"{5 SPACES}HOUR
     (0-23) ";T1:IFT1<0ORT1>23
    THENPRINTOO$:GOTO400
                         :rem 72
410 PRINT:INPUT"{3 SPACES}MINU
    TE (0-59) ";T2:IFT2<0ORT2>
    59THENPRINTOO$:GOTO410
                         :rem 243
420 R$=STR$(T1):T$=STR$(T2):T$
    =RIGHT$(T$,LEN(T$)-1):IFLE
    N(T$)=1THENT$="0"+T$
                         :rem 133
430 PRINT"{2 DOWN}TIME-- "R$":
    "T$                  :rem 127
440 PRINT:GOSUB2230:IFZ$="N"TH
    EN380                :rem 134
450 PRINT"{CLR}":T3=T1*60+T2+A
    A-720:IFT3<0THENT3=T3+K1
                         :rem 17
460 IFT3>K1THENT3=T3-K1
                         :rem 141
470 MT=T3-360:IFMT<0THENMT=MT+
    K1                   :rem 241
480 PT=T3+360:IFPT>K1THENPT=PT
    -K1                  :rem 76
490 GOSUB1770:PRINTTAB(27)R$":
    "T$                  :rem 176
500 C9$="{BLU}":TM=VAL(R$+"."+
    T$):IFTM<6ORTM>18THENC9$="
    {BLK}"               :rem 124
510 XX=7+LC:FORX=1TO14:IFX=XXT
    HEN530               :rem 56
520 PRINTC9$+"{RVS}{40 SPACES}
    ";"{BLK}";:GOTO540:rem 155
530 PRINTC9$+"{RVS} - - - - -
    {SPACE}- - - - - - - - - -
     - - - -{2 SPACES}";"{BLK}
    ";                   :rem 231
540 NEXTX:GOSUB340:IFLL<0THEN5
    70                   :rem 25
550 IFLL>24THENPRINT"{BLU}E"SP
    C(18)"S"SPC(19)"W{BLK}":GO
    TO590                :rem 221
560 PRINT"{BLU}UP-{BLK}NORTH
    {5 SPACES}{BLU}----{BLK}OV
    ERHEAD{5 SPACES}{BLU}DOWN-
    {BLK}SOUTH":GOTO590
                         :rem 225
570 IFABS(LL)>24THENPRINT"
    {BLU}W"SPC(18)"N"SPC(19)"E
    {BLK}":GOTO590       :rem 1
580 PRINT"{BLU}UP-{BLK}SOUTH
    {5 SPACES}{BLU}----{BLK}OV
    ERHEAD{5 SPACES}{BLU}DOWN-
    {BLK}NORTH"          :rem 210
590 T4=AA:GOSUB800:Y8=888:IFY9
    =999THEN630          :rem 242
600 Y8=Y9:GOSUB2450:IFA1<0THEN
    630                  :rem 234
610 IFPK>1703ORPK<1144THEN630
                         :rem 212
620 POKEPK,170           :rem 38
630 T4=AA+M2*K1:IFT4>K1THENT4=
    T4-K1                :rem 96
640 GOSUB800:IFY9=999THEN680
                         :rem 194
650 MM=INT(M1/9.83333)+1:GOSUB
    900:IFY9=999THEN680
                         :rem 133
660 GOSUB2450:IFPK>1703ORPK<11
    44THEN680            :rem 99
670 POKEPK,MM+128:PRINTCHR$(14
    6):IFABS(Y8-Y9)<=.5THENPOK
    EPK,81               :rem 81
680 FOR X=1TO7:IFX=7THEN2350
                         :rem 179
690 T4=P(X,6):GOSUB800:IFY9=99
    9THEN750             :rem 31
700 U9=SIN(P(X,6)*D9/4):U9=-3*
    U9+.5:U9=INT(U9):U(X)=U9*4
    0                    :rem 13
710 PK=1423-Y9+U(X)+LB:GOSUB24
    60                   :rem 97
720 IFPK>1703ORPK<1144THEN750
                         :rem 217
730 Z=PEEK(PK):IFZ<>160ANDZ<>1
    73THENPK=PK+SGN(LL)*40+(LL
    =0)*40:GOTO730       :rem 0
740 POKEPK,PP(X)         :rem 218
750 NEXTX:PRINT"{HOME}
    {19 DOWN}"           :rem 148
760 PRINT"{UP}VMERCURY
    {2 SPACES}WVENUS{4 SPACES}
    XMARS{5 SPACES}YJUPITER"
                         :rem 107
770 PRINT"ZSATURN{4 SPACES}+UR
    ANUS{3 SPACES}{RVS}*{OFF}S
    UN{6 SPACES}{RVS})Q({OFF}M
    OON"                 :rem 162
780 PRINT"{2 SPACES}QNEW MOON
    {SPACE}+ SUN{2 SPACES}"B$
                         :rem 235
790 PRINT:PRINT"T- NEW TIME,P-
     P.TABLE,D- DATE,L- LAT":G
    OTO1920              :rem 225
800 Y9=999:IFMT<PTTHEN850
                         :rem 40
810 IFT4>=MT OR T4<=PTTHEN830
                         :rem 220
820 RETURN               :rem 122
830 IFT4>=MT AND T4<=K1THEN870
                         :rem 236
840 T4=T4+K1:GOTO870     :rem 162
850 IFT4>=MT AND T4<=PT THEN87
    0                    :rem 22
860 RETURN               :rem 126
870 Y9=INT((T4-MT)/18+.5):IFY9
    =40THENY9=39         :rem 221
880 RETURN               :rem 128
890 U9=SIN(T4/4*D9):U9=-3*U9+.
    5:U9=INT(U9):U9=U9*40:RETU
    RN                   :rem 238
900 MM=VAL(MID$(MM$,3*MM-2,3))
    :IFLL<0ANDMM<>81THENMM=ABS
    (MM-81)              :rem 12
910 RETURN               :rem 122
920 PRINT"{CLR}{DOWN}
    {6 SPACES}********** SKYSC
    APE **********":PRINT"
    {DOWN}DATE INPUT":S1=0
                         :rem 176
930 PRINT"E10 Y3":IFY<>0THENGO
    SUB1770:PRINT:PRINT
                         :rem 107
940 INPUT"YEAR{2 SPACES}";Y:IF
    Y<1977THENPRINT"MUST BE GR
    EATER THAN 1977":GOTO940
                         :rem 89
950 GOSUB1820:PRINT:INPUT"MONT
    H (1-12) ";M:IFM<1ORM>12TH
    ENPRINTOO$:GOTO950:rem 127
960 DI=VAL(MID$(MD$,2*M-1,2)):
    DI=DI-(M=2)*LY:DI$=STR$(DI
    ):DI$=RIGHT$(DI$,2):rem 25
970 PRINT"{DOWN}DAY (1-"DI$")
    {SPACE}";:INPUTD:IFD<1ORD>
    DITHENPRINTOO$:GOTO970
                         :rem 8
980 H$=MID$(A$,M*3-2,3):PRINT:
    PRINT"LATITUDE (-90 TO 90)
    ";:INPUTLL           :rem 80
990 GOSUB2480            :rem 240
1000 IFABS(LL)>90THENPRINTOO$:
     GOTO980             :rem 72
1010 PRINT:PRINT"{2 DOWN}
     {4 RIGHT}"H$;D"{LEFT},"Y:
     PRINT:GOSUB2230:IF Z$="N"
     THEN920             :rem 105
```


It's easy to make a copy.
It's quick.
It's illegal.
It's wrong.
It's hard to believe.
People who wouldn't think of shoplifting a software product on their lunch hour don't think twice about going back to the office and making several illegal copies of the same software.
Making unauthorized copies of software is a violation of U.S. Copyright Law. Yet, the problem has reached epidemic proportions because many people are unaware, or simply choose to ignore the law. The software industry is urging decision-makers and software users to take steps to stop software piracy in their organizations. In the meantime, the industry has been forced to prosecute willful copyright violators.
There are legal, moral and economic imperatives forbidding theft of copyrighted software.
There is a free pamphlet on the subject. Call or write for a copy. A copy. A copy. A copy for everyone you know. Please ask for Priscilla.
ADAPSO
ADAPSO
1300 North Seventeenth Street
Arlington, Virginia 22209
(703) 522-5055

```
1020 D2=VAL(MID$(M$,M*3-2,3))+D:GOSUB1860:IFM>2THEND1=D1+LY:Y1=Y1+LY :rem 253
1030 D3=D2-185:IFM=3ANDD<20THEND2=D2+LY:D3=D3+LY:rem 81
1040 S=Ø:IFD3<=ØTHENA=180*D2/185:GOTO1060 :rem 91
1050 A=180*D3/(180+ZY)+180 :rem 57
1060 IFA<>180THENS=23.43333333*SIN(D9*D2*180/185) :rem 167
1070 IFA>180THENS=-23.43333333*SIN(D9*D3) :rem 10
1080 IFA>=360THENA=A-360 :rem 97
1090 A=FNR(A) :rem 192
1100 S=FNR(S):A1=(SGN(LL)-(LL=Ø))*S+90-ABS(LL):A1=FNR(A1):GOSUB1470:GOSUB1420 :rem Ø
1110 W=1-(SGN(LL)<>Ø):IFA1>90THENA1=180-A1:W=ABS(W-3) :rem 231
1120 PRINT"{CLR}{DOWN}":GOSUB1770:PRINT:PRINT"{32 Y}":I$="{LEFT}@" :rem 121
1130 PRINT:PRINT"DAY OF THE YEAR-----------",D1:rem 114
1140 PRINT"SUN'S GEOCENTRIC ANGLE----",A;I$ :rem 26
1150 PRINT"SUN'S DECLINATION---------",S;I$ :rem 238
1160 PRINT"SUN'S ALTITUDE AT NOON----",A1;I$;D$(W) :rem 121
1170 PRINT"SUN'S RIGHT ASCENSION-----",A3$ :rem 208
1180 PRINT"R.A. AT 9:ØØPM-----------",A5$ :rem 113
1190 PRINT"MOON'S AGE----------------",M1;"DY" :rem 178
1200 PRINT"MOON'S ELONGATION---------",M8;"{LEFT}@";L$:I$="" :rem 172
1210 PRINT"MOON'S PHASE - "PH$(M3) :rem 115
1220 PRINT"{2 DOWN}-P- PLANET{SPACE}TABLE , -D- NEW DATE":GOTO1920 :rem 159
1230 PRINT"{CLR}{DOWN}":PRINTTAB(7)"** PLANET TABLE **":GOSUB1770:PRINT:PRINT:S1=1 :rem 188
1240 PRINT"PLANET{3 SPACES}DIST.{2 SPACES}ANG. W/ SUN{4 SPACES}R.A" :rem 255
1250 PRINT"{38 Y}":PRINT :rem 138
1260 FORX=1TO6:A2=Y1/P(X,2)-INT(Y1/P(X,2)):Q3=1:rem 238
1270 A2=A2*360+P(X,1):IFA2>360THENA2=A2-360 :rem 92
1280 E=180+A:IFE>360THENE=E-360 :rem 243
1290 E1=ABS(E-A2):IFE1>180THENE1=360-E1 :rem 191
1300 GOSUB1530:E1=E1*D9:P5=P(X,3):IFX=3THENGOSUB1980 :rem 125
1310 P(X,4)=SQR(1+P5↑2-2*P5*COS(E1)):XX=(P5↑2-1-P(X,4)↑2)/(-2*P(X,4)) :rem 90
1320 P(X,5)=-ATN(XX/SQR(-XX*XX+1))+↑/2:P(X,4)=INT(P(X,4)*ES+.5) :rem 55
1330 P(X,5)=P(X,5)/D9:P(X,5)=FNS(P(X,5)):Q1$=STR$(P(X,4)):Q2$=STR$(P(X,5)) :rem 145
1340 Q1=LEN(Q1$):Q2=LEN(Q2$):GOSUB1630 :rem 178
1350 PRINTP$(X);TAB(14-Q1);Q1$;TAB(22-Q2);Q2$;:IFQ3=-1THENPRINT"@W"; :rem 25
1360 IFQ3=1THENPRINT"@E"; :rem 11
1370 GOSUB1680:Q4$=STR$(Q4):Q5$=STR$(Q5):IFQ5<10THENQ5$="Ø"+RIGHT$(Q5$,1) :rem 221
1380 Q5$=RIGHT$(Q5$,2):Q4$=Q4$+":"+Q5$:Z=LEN(Q4$) :rem 159
1390 PRINTTAB(26)QQ$TAB(34-Z)Q4$:NEXT:PRINT"{2 DOWN}* - VISIBLE AT 9 P.M." :rem 65
1400 PRINT"{2 DOWN}SUN'S R.A.{SPACE}-------"SPC(Q8)A3$:PRINT"R.A. AT 9:ØØPM ---"SPC(Q9)A5$ :rem 139
1410 PRINT"{DOWN}-S- FOR DAYS{SPACE}SKY -D- FOR NEW DATE":GOTO1920 :rem 48
1420 A2=K1*A/360:IFA2>K1THENA2=A2-K1 :rem 23
1430 A3=INT(A2/60):A4=A2-A3*60:A5=A3+9:IFA5>23THENA5=A5-24 :rem 223
1440 A4=INT(A2-A3*60+.5):IFA4=60THENA4=Ø:A3=A3+1 :rem 150
1450 IFA3=24THENA3=Ø :rem 128
1460 AA=A3*60+A4:GOTO1780 :rem 113
1470 M1=(Y1/M9-INT(Y1/M9))*M9+10:IFM1>M9THENM1=M1-M9 :rem 33
1480 GOSUB2260:M8=360*M2:IFM8>180THENL$="W" :rem 241
1490 IFM8<=180THENL$="E" :rem 89
1500 IFM8>180THENM8=360-M8 :rem 237
1510 M1=FNR(M1):M8=FNR(M8):YY=INT(7*(Y1/7-INT(Y1/7))+.2):IFYY=ØTHENYY=7 :rem 23
1520 K$=MID$(J$,YY*3-2,3):RETURN :rem 68
1530 Q3=Ø:Q1=E+180:IFQ1>360THEN1570 :rem 218
1540 IFA2>EANDA2<Q1THEN1560 :rem 78
1550 Q3=1:RETURN :rem 215
1560 Q3=-1:RETURN :rem 5
1570 Q1=Q1-360:IFA2<=360ANDA2>ETHEN1560 :rem 230
1580 IFQ3<>ØTHENRETURN:rem 154
1590 IFA2>ØANDA2<=Q1THEN1560 :rem 123
1600 IFQ3<>ØTHENRETURN:rem 147
1610 IFA2>Q1THEN1550 :rem 132
1620 RETURN :rem 169
1630 Q5=Q3*P(X,5)*4+AA:IFQ5<ØTHENQ5=Q5+K1 :rem 122
1640 IFQ5>K1THENQ5=Q5-K1 :rem 187
1650 P(X,6)=Q5:Q4=INT(Q5/60):Q5=INT(Q5-Q4*60+.5):IFQ5=6ØTHENQ5=Ø:Q4=Q4+1:rem 189
1660 IFQ4=24THENQ4=Ø :rem 165
1670 RETURN :rem 174
1680 SU=A5*60+A4:PS=SU+360:MS=SU-360:IFPS>K1THENPS=PS-K1 :rem 230
1690 IFMS<ØTHENMS=MS+K1 :rem 190
1700 IF MS>PSTHEN1730 :rem 210
1710 IFP(X,6)<PSANDP(X,6)>MSTHEN1760 :rem 155
1720 QQ$=" ":RETURN :rem 43
1730 IF P(X,6)<K1ANDP(X,6)>MSTHEN1760 :rem 118
1740 IFP(X,6)<PSTHEN1760 :rem 146
1750 GOTO1720 :rem 208
1760 QQ$="*":RETURN :rem 89
1770 PRINT:PRINT K$"-- "H$;D"{LEFT},"Y;TAB(20)ABS(LL)LL$;:RETURN :rem 22
1780 A3$=STR$(A3):A3$=RIGHT$(A3$,2):A4$=STR$(A4):A4$=RIGHT$(A4$,2) :rem 108
1790 IFA4<10THENA4$="Ø"+RIGHT$(A4$,1) :rem 255
1800 A3$=A3$+":"+RIGHT$(A4$,2):A5$=STR$(A5):A5$=RIGHT$(A5$,2)+":"+A4$ :rem 82
1810 Q8=7-LEN(A3$):Q9=7-LEN(A5$):RETURN :rem 5
1820 LY=Ø:IFY/4=INT(Y/4)THENLY=1 :rem 217
1830 IFY/100=INT(Y/100)ANDY/400<>INT(Y/400)THENLY=Ø :rem 8
1840 IFY/1000=INT(Y/1000)ANDY/4000=INT(Y/4000)THENLY=Ø :rem 140
1850 RETURN :rem 174
1860 Y9=Y+1:IFY9/4=INT(Y9/4)THENZY=1 :rem 207
1870 IFY9/100=INT(Y9/100)ANDY9/400<>INT(Y9/400)THENZY=Ø :rem 254
1880 IFY9/1000=INT(Y9/1000)ANDY9/4000=INT(Y9/4000)THENZY=Ø :rem 130
1890 Y1=Y-1977:Y1=Y1*365+INT(Y1/4)+D1:IFY<2000THEN1910 :rem 88
1900 Y1=Y1-INT((Y-2001)/100)+INT((Y-2001)/400)-INT((Y-1)/4000) :rem 6
1910 RETURN :rem 171
1920 GETI$:IFI$=""THEN1920 :rem 203
1930 IFI$="D"THEN920 :rem 88
1940 IF(I$="S"ORI$="T")ANDS1=1THEN380 :rem 97
1950 IFI$="P"THEN1230 :rem 145
1960 IFI$="L"ANDS1=1THEN2530 :rem 87
1970 GOTO1920 :rem 214
1980 P5=1.376344086:K5=A2*4 :rem 148
1990 K5=ABS(K5-1233.73)*90/K1:K5=K5*D9:K5=SIN(K5)*.322581224:P5=P5+K5:RETURN :rem 62
2000 IFCC<=1THENCC=CC+84 :rem 144
2010 CD$=MID$(CC$,CC-1,42) :rem 70
2020 IFMID$(CD$,2,1)<>" "ANDMID$(CD$,3,1)=" "THENCD$=MID$(CD$,1,40):GOTO2050 :rem 8
2030 IFMID$(CD$,41,1)<>" "ANDMID$(CD$,40,1)=" "THENCD$=MID$(CD$,3,42):GOTO2050 :rem 113
2040 CD$=MID$(CD$,2,40) :rem 150
2050 CD$="{YEL}"+CD$+"{BLK}":RETURN :rem 128
2060 DATA365.26,29.53059,59.818184,42.719626,262.364294,52.916763 :rem 23
2070 DATA134.69697,218.79464,87.97,224.7,686.98:rem 146
2080 DATA4332.79813,10759.7195,30686.5884 :rem 90
2090 DATA"MERCURY",.3871,"VENUS",.7233,"MARS",1.5237,"JUPITER",5.2028 :rem 148
```

```
2100 DATA"SATURN",9.5308,"URAN
     US",19.182         :rem 14
2110 DATA56,68,68,68,56,0,0,0,
     255,239,199,131,131,199,2
     39,255              :rem 6
2120 DATA195,129,153,153,195,2
     31,129,231,252,249,195,15
     3,153,153,199,255  :rem 40
2130 DATA255,195,189,129,129,1
     89,195,255,252,193,145,13
     7,153,131,63,127    :rem 2
2140 DATA255,153,153,153,219,2
     31,255,255         :rem 230
2150 DATA245,234,213,202,213,1
     39,7,31            :rem 66
2160 DATA"SA","SC","LI","VI","
     LE","CA","GE","TA","AR","
     PI","AQ","CP"      :rem 220
2170 DATA"NEW","WAXING CRESCEN
     T","1ST QUARTER","WAXING
     {SPACE}GIBBOUS","FULL"
                        :rem 255
2180 DATA"WANING GIBBOUS","3RD
      QUARTER","WANING CRESCEN
     T"                 :rem 224
2190 DATA1770,1719,1620,1500,1
     418,1365,1335,1310,1290,1
     275,1260           :rem 96
2200 DATA1238,1220,1200,1178,1
     115,915,720,660,640,625,6
     10                 :rem 39
2210 PRINT"{CLR}{11 DOWN}"SPC(
     11)"**** SKYSCAPE ****"
                        :rem 116
2220 POKE53272,(PEEK(53272)AND
     240)OR14:SYS828:POKE53280
     ,7:RETURN          :rem 32
2230 PRINT"-N- TO RE-INPUT OR
     {SPACE}RETURN TO CONTINUE
     "                  :rem 192
2240 GETZ$:IFZ$=""THEN2240
                        :rem 229
2250 RETURN             :rem 169
2260 M2=M1/M9:IFM1<1ORM1>28.5T
     HENM3=1            :rem 180
2270 IFM1>=1ANDM1<6.9THENM3=2
                        :rem 204
2280 IFM1<=8.0ANDM1>=6.9THENM3
     =3                 :rem 112
2290 IFM1>8.0ANDM1<14.2THENM3=
     4                  :rem 32
2300 IFM1>=14.2ANDM1<=15.2THEN
     M3=5               :rem 195
2310 IFM1>15.2ANDM1<21.6THENM3
     =6                 :rem 77
2320 IFM1>=21.6ANDM1<=22.6THEN
     M3=7               :rem 203
2330 IFM1>22.6ANDM1<=28.5THENM
     3=8                :rem 150
2340 RETURN             :rem 169
2350 B$="":IFY<>1985ANDY<>1986
     THEN750            :rem 109
2360 IF(Y=1985ANDD1<305)OR(Y=1
     986ANDD1>149)THEN750
                        :rem 131
2370 HD=D1+365:IFHD>516THENHD=
     HD-365             :rem 81
2380 H1=(HD-295)/10:HD=INT(H1)
     :H1=H1-HD          :rem 151
2390 T4=HC(HD)-HC(HD+1):T4=HC(
     HD)-H1*T4:IFT4>K1THENT4=T
     4-K1               :rem 219
2400 GOSUB800:IFY9=999THEN750
                        :rem 236
2410 GOSUB890:IFT4>1115ANDT4<1
     200THENU9=U9+40    :rem 176
2420 IFT4>1290THENU9=U9-40
                        :rem 2
2430 IFT4>615ANDT4<=1115THENU9
     =U9+80             :rem 113
2440 U(7)=U9:B$="E-3HALLEY'S C
     OMET":GOTO710      :rem 43
2450 GOSUB890:PK=1423-Y9+U9+LB
                        :rem 249
2460 IFLL<0THENPK=2247+80*XX-P
     K                  :rem 106
2470 RETURN             :rem 173
2480 LL$="{LEFT}@N":IFLL<0THEN
     LL$="{LEFT}@S"     :rem 159
2490 L1=ABS(LL):IFABS(LL)<24TH
     ENL1=40            :rem 191
2500 LC=INT((L1-40)/7+.5):LB=L
     C*40:D1=VAL(MID$(D$,M*3-2
     ,3))+D             :rem 30
2510 IFABS(LL)<24THENLB=40*INT
     (ABS(LL)/7+.5)     :rem 47
2520 RETURN             :rem 169
2530 PRINT"{CLR}{DOWN}
     {6 SPACES}********** SKYS
     CAPE **********":PRINT"
     {DOWN}LATITUDE CHANGE"
                        :rem 8
2540 PRINT"E15 Y3":GOSUB 1770
                        :rem 222
2550 PRINT"{3 DOWN}":INPUT"INP
     UT NEW LATITUDE";LL:PRINT
     :PRINT             :rem 21
2560 IF ABS(LL)>90 THEN PRINT
     {SPACE}OO$:GOTO 2550
                        :rem 127
2570 GOSUB2230:IFZ$="N"THEN248
     0                  :rem 40
2580 GOSUB2480:I$="S":GOTO1940
                        :rem 217
2590 FORX=1704TO1723:U1=PEEK(X
     ):U2=PEEK(3447-X):POKEX,U
     2:POKE(3447-X),U1:NEXT
                        :rem 206
2600 FORX=1704TO1742:IFPEEK(X)
     =160THEN2620       :rem 229
2610 U1=PEEK(X):U2=PEEK(X+1):P
     OKEX,U2:POKEX+1,U1:X=X+1
                        :rem 72
2620 NEXT:RETURN        :rem 35
```



A view of the night sky in the Atari version of "Skyscape."

Program 2: Atari Skyscape

Version by Kevin Mykytyn, Editorial Programmer

```
CG 100 POKE 106,PEEK(106)-5:
       GRAPHICS 0:OPEN #1,4,
       0,"K:"
HO 110 DIM D$(36),MM$(9),M$(
       36),A$(36),OO$(16),MD
       $(24),ZZ$(20),P$(43),
       J$(21),F$(200),CC$(20
       0),SPC$(30),Q$(1)
BD 120 DIM PH$(120),HC(22),R
       $(10),T$(10),Q1$(10),
       Q2$(10),Q3$(10),Q4$(1
       0),Q5$(10),L$(10),QQ$
       (10),A1$(10),A2$(10),
       A4$(10)
CM 130 DIM A3$(10),I$(2),CD$
       (50),PP(8),DI$(3),H$(
       5),LL$(3),DIR$(2),Z$(
       1),K$(21),A5$(10),U(1
       0),P(6,6),B$(15)
KI 140 FOR A=1 TO 6:FOR B=1
       TO 6:P(A,B)=0:U(A)=0:
       NEXT B:NEXT A
GN 150 POKE 82,0:PI=3.1415:S
       CREEN=PEEK(88)+256*PE
       EK(89):FOR X=1 TO 30:
       SPC$(X,X)=" ":NEXT X
EB 160 P$=" ":P$(43)=P$:P$(2
       )=P$:PH$=" ":PH$(120)
       =PH$:PH$(2)=PH$
NO 170 GOSUB 2220
JO 180 D$="000031059090120l5
       1181212243273304334":
       K1=1440:MM$="0090B400
       B"
BD 190 M$="2B631734501104107
       2102133164194225255":
       DIR$(1,1)="S":DIR$(2,
       2)="N":ES=93
PN 200 A$="JANFEBMARAPRMAYJU
       NJULAUGSEPOCTNOVDEC":
       OO$="{DOWN}OUT OF RAN
       GE''{DOWN}"
PO 210 MO$="3128313031303131
       30313031":D9=3.141592
       65/180:READ EE:READ M
       9:GOTO 240
MI 220 ZZ=INT(ZZ*100+0.5)/10
       0:RETURN
GJ 230 ZZ=INT(ZZ*10+0.5)/10:
       RETURN
CP 240 FOR Y=1 TO 2:FOR X=1
       TO 6:READ ZZ:P(X,Y)=Z
       Z:NEXT X:NEXT Y:Y=0
LP 250 FOR X=1 TO 6:READ ZZ$
       :P$((X-1)*7+1,X*7)=ZZ
       $:READ ZZ:P(X,3)=ZZ:N
       EXT X:FOR X=1 TO 8:RE
       AD A
LB 260 POKE CHBAS+256+X,A:NE
       XT X
MN 270 FOR X=CHBAS+608 TO CH
       BAS+663:READ A:POKE X
       ,255-A:POKE X+1024,A:
       NEXT X:FOR X=1 TO 7:P
       P(X)=X+75:NEXT X
JO 280 J$="SATSUNMONTUEWEDTH
       UFRI":FOR X=1 TO 12:R
       EAD F$
PO 290 CC$((X-1)*7+1,(X-1)*7
       +5)="{5 SPACES}":CC$(
       (X-1)*7+6,X*7)=F$:NEX
       T X:CC$(LEN(CC$)+1,2*
       LEN(CC$))=CC$
KC 300 F$=CC$(LEN(CC$)-8,LEN
       (CC$)):F$(LEN(F$)+1,L
       EN(F$)+LEN(CC$))=CC$:
       CC$=F$
HO 310 FOR X=1 TO 8:READ ZZ$
       :PH$((X-1)*15+1,X*15)
       =ZZ$:NEXT X
NN 320 FOR X=1 TO 22:READ ZZ
       :HC(X)=ZZ:NEXT X:FOR
       X=CHBAS+680 TO CHBAS+
       687:READ B:POKE X,B:N
       EXT X:GOTO 920
JK 330 CC=MT-720:IF CC<0 THE
       N CC=CC+K1
CO 340 CC=CC/120:CD=CC-INT(C
       C):CC=INT(CC):CD=INT(
       CD*7+0.2):CC=81-(CC*7
       +CD)
MO 350 GOSUB 2010:PRINT CD$;
       :IF LL<0 THEN GOSUB 2
       620
HJ 360 RETURN
IA 370 PRINT "{CLEAR}":POSIT
       ION 10,1:PRINT "** DA
       YS SKY **":GOSUB 1770
       :PRINT
```

```
BJ 38Ø R$="":T$="":ZZ$="":PR
       INT :PRINT "INPUT THE
       TIME ":PRINT "{15 U}
       ":T1=Ø:T2=Ø
BP 39Ø PRINT :PRINT "
       {5 SPACES}HOUR (Ø-23)
       ";:INPUT T1:IF T1<Ø
       OR T1>23 THEN PRINT O
       O$:GOTO 39Ø
LI 4ØØ PRINT :PRINT "
       {3 SPACES}MINUTE (Ø-5
       9) ";:INPUT T2:IF T2<
       Ø OR T2>59 THEN PRINT
        OO$:GOTO 4ØØ
PB 41Ø R$=STR$(T1):T$=STR$(T
       2):IF LEN(T$)=1 THEN
       ZZ$="Ø":ZZ$(2,LEN(T$)
       +1)=T$:T$=ZZ$
EH 42Ø PRINT "{2 DOWN}TIME--
       ";R$;":";T$
IH 43Ø PRINT :GOSUB 226Ø:IF
       Z$="N" THEN 37Ø
PK 44Ø PRINT "{CLEAR}":T3=T1
       *6Ø+T2+AA-72Ø:IF T3<Ø
        THEN T3=T3+K1
IM 45Ø IF T3>K1 THEN T3=T3-K
       1
PA 46Ø MT=T3-36Ø:IF MT<Ø THE
       N MT=MT+K1
EL 47Ø PT=T3+36Ø:IF PT>K1 TH
       EN PT=PT-K1
LK 48Ø GOSUB 177Ø:PRINT SPC$
       (1,3);R$;":";T$
NP 49Ø RF=128:ZZ$=R$:ZZ$(LEN
       (ZZ$)+1,LEN(ZZ$)+1)="
       .":ZZ$(LEN(ZZ$)+1,LEN
       (ZZ$)+LEN(T$))=T$:TM=
       VAL(ZZ$)
FE 5ØØ IF TM<6 OR TM>18 THEN
        RF=Ø
DI 51Ø XX=7+LC:FOR X=1 TO 14
       :IF X=XX THEN 53Ø
HG 52Ø FOR A=1 TO 4Ø:PRINT C
       HR$(RF+32);:NEXT A:GO
       TO 54Ø
FF 53Ø FOR ZZ=1 TO 4Ø:PRINT
       CHR$(45+RF);:NEXT ZZ
BI 54Ø NEXT X:GOSUB 33Ø:IF L
       L<Ø THEN 57Ø
GF 55Ø IF LL>24 THEN PRINT "
       E{18 SPACES}S
       {19 SPACES}W":GOTO 59Ø
NE 56Ø PRINT "UP-NORTH
       {5 SPACES}----OVERHEA
       D{5 SPACES}DOWN-SOUTH
       ":GOTO 59Ø
IJ 57Ø IF ABS(LL)>24 THEN PR
       INT "W{18 SPACES}N
       {19 SPACES}E":GOTO 59Ø
NF 58Ø PRINT "UP-SOUTH
       {5 SPACES}----OVERHEA
       D{5 SPACES}DOWN-NORTH
       "
PC 59Ø T4=AA:GOSUB 8ØØ:Y8=8B
       8:IF Y9=999 THEN 63Ø
DN 6ØØ YB=Y9:GOSUB 248Ø:IF A
       1<Ø THEN 63Ø
ED 61Ø IF PK>SCREEN+679 OR P
       K<SCREEN+12Ø THEN 63Ø
LC 62Ø POKE PK,1Ø+RF
GA 63Ø T4=AA+M2*K1:IF T4>K1
       THEN T4=T4-K1
MC 64Ø GOSUB 8ØØ:IF Y9=999 T
       HEN 68Ø
IF 65Ø MM=INT(M1/9.83333)+1:
       GOSUB 9ØØ:IF Y9=999 T
       HEN 68Ø
OA 66Ø GOSUB 248Ø:IF PK>SCRE
       EN+679 OR PK<SCREEN+1
       2Ø THEN 68Ø
MN 67Ø POKE PK,MM+RF:IF ABS(
       Y8-Y9)<=Ø.5 THEN POKE
        PK,84
LG 68Ø FOR X=1 TO 7:IF X=7 T
       HEN 238Ø
BP 69Ø T4=P(X,6):GOSUB 8ØØ:I
       F Y9=999 THEN 75Ø
DN 7ØØ U9=SIN(P(X,6)/4*O9):U
       9=-3*U9+Ø.5:U9=INT(U9
       ):U(X)=U9*4Ø
CK 71Ø PK=SCREEN+399-Y9+U(X)
       +L8:GOSUB 249Ø
FD 72Ø IF PK>SCREEN+679 OR P
       K<SCREEN+12Ø THEN 75Ø
NL 73Ø Z=PEEK(PK):IF Z<>RF A
       ND Z<>13+RF THEN PK=P
       K+SGN(LL)*4Ø-(LL=Ø)*4
       Ø:PRINT "A":GOTO 73Ø
JN 74Ø POKE PK,PP(X)+RF
KI 75Ø NEXT X:POSITION Ø,19
LC 76Ø PRINT "{■}MERCURY
       {■}VENUS{4 SPACES}
       {■}MARS{5 SPACES}{■}J
       UPITER"
AJ 77Ø PRINT "{■}SATURN
       {4 SPACES}{■}URANUS
       {3 SPACES}■SUN
       {6 SPACES}■{■}■MOON"
GJ 78Ø PRINT "  {T}NEW MOON
       +SUN  ";8$
BN 79Ø PRINT :PRINT "T- NEW
       TIME,P- P.TABLE,D- DA
       TE,L- LAT";:GOTO 193Ø
CI 8ØØ Y9=999:IF MT<PT THEN
       85Ø
NM 81Ø IF T4>=MT OR T4<=PT T
       HEN 83Ø
HK 82Ø RETURN
DM 83Ø IF T4>=MT AND T4<=K1
       THEN 87Ø
KC 84Ø T4=T4+K1:GOTO 87Ø
BG 85Ø IF T4>=MT AND T4<=PT
       THEN 87Ø
HO 86Ø RETURN
AN 87Ø Y9=INT((T4-MT)/18+Ø.5
       ):IF Y9=4Ø THEN Y9=39
IA 88Ø RETURN
NE 89Ø U9=SIN(T4/4/(1/O9)):U
       9=-3*U9+Ø.5:U9=INT(U9
       ):U9=U9*4Ø:RETURN
KF 9ØØ MM=VAL(MM$(3*MM-2,3*M
       M)):IF LL<Ø AND MM<>8
       1 THEN MM=ABS(MM-17)
HK 91Ø RETURN
NM 92Ø PRINT "{CLEAR}{DOWN}
       {5 SPACES}**********
       SKYSCAPE ***********"
       :PRINT "{DOWN}DATE IN
       PUT":S1=Ø
BH 93Ø PRINT "{1Ø U}":IF Y<>
       Ø THEN GOSUB 177Ø:PRI
       NT :PRINT
CA 94Ø PRINT "YEAR  ";:INPUT
        Y:IF Y<1977 THEN PRI
       NT "MUST BE GREATER T
       HAN 1977":GOTO 94Ø
EH 95Ø GOSUB 183Ø:PRINT :PRI
       NT "MONTH (1-12) ";:I
       NPUT M:IF M<1 OR M>12
        THEN PRINT OO$:GOTO
       95Ø
DK 96Ø DI=VAL(MO$(2*M-1,2*M)
       ):DI=DI+(M=2)*LY:DI$=
       STR$(DI)
IK 97Ø PRINT "{DOWN}DAY (1-"
       ;DI$;") ";:INPUT D:IF
        D<1 OR D>DI THEN PRI
       NT OO$:GOTO 97Ø
JN 98Ø H$=A$(M*3-2,M*3):PRIN
       T :PRINT "LATITUDE (-
       9Ø TO 9Ø)";:INPUT LL
OK 99Ø GOSUB 251Ø
EG 1ØØØ IF ABS(LL)>9Ø THEN P
        RINT OO$:GOTO 78Ø
BP 1Ø1Ø PRINT :PRINT "
        {2 DOWN}{4 RIGHT}";H
        $;" ";D;",";Y:PRINT
        :GOSUB 226Ø:IF Z$="N
        " THEN 92Ø
JM 1Ø2Ø D2=VAL(M$((M*3)-2,M*
        3))+D:GOSUB 187Ø:IF
        M>2 THEN D1=D1+LY:Y1
        =Y1+LY
FB 1Ø3Ø D3=D2-185:IF M=3 AND
         D<2Ø THEN D2=D2+LY:
        D3=D3+LY
FL 1Ø4Ø S=Ø:IF D3<=Ø THEN A=
        18Ø*D2/185:GOTO 1Ø6Ø
IK 1Ø5Ø A=(18Ø*D3/(18Ø+ZY))+
        18Ø
EB 1Ø6Ø IF A<>18Ø THEN S=23.
        433333*SIN(O9*D2*18Ø
        /185)
NH 1Ø7Ø IF A>18Ø THEN S=-23.
        4333333*SIN(O9*D3)
GB 1Ø8Ø IF A>=36Ø THEN A=A-3
        6Ø
LG 1Ø9Ø ZZ=A:GOSUB 22Ø:A=ZZ
DK 11ØØ ZZ=S:GOSUB 22Ø:S=ZZ:
        A1=(SGN(LL)+(LL=Ø))*
        S+9Ø-ABS(LL):ZZ=A1:G
        OSUB 22Ø:A1=ZZ:GOSUB
         147Ø:GOSUB 142Ø
OG 111Ø W=1+(SGN(LL)<Ø):IF A
        1>9Ø THEN A1=18Ø-A2:
        W=ABS(W-3)
JC 112Ø PRINT "{CLEAR}{DOWN}
        ":GOSUB 177Ø:PRINT :
        PRINT "{32 U}":I$="@
        "
HC 113Ø PRINT :PRINT "DAY OF
         THE YEAR-----------
        ",D1
BK 114Ø PRINT "SUN'S GEOCENT
        RIC ANGLE----",A;I$
OO 115Ø PRINT "SUN'S DECLINA
        TION---------",S;I$
JH 116Ø PRINT "SUN'S ALTITUD
        E AT NOON----",A1;I$
        ;DIR$(W,W)
NA 117Ø PRINT "SUN'S RIGHT A
        SCENSION-----",A3$
HB 118Ø PRINT "R.A. AT 9:ØØP
        M------------",A5$
LC 119Ø PRINT "MOON'S AGE---
        --------------",M1;"
        DY"
PI 12ØØ PRINT "MOON'S ELONGA
        TION---------",M8;I$
        ;L$:I$=""
IF 121Ø PRINT "MOON'S PHASE
        - ";PH$((M3-1)*15+1,
        M3*15)
LI 122Ø PRINT "{2 DOWN}-P- P
        LANET TABLE , -D- NE
        W DATE":GOTO 193Ø
FD 123Ø PRINT "{CLEAR}{DOWN}
        ":PRINT "{7 SPACES}*
        * PLANET TABLE **":G
        OSUB 177Ø:PRINT :PRI
        NT :S1=1
PP 124Ø PRINT "PLANET
        {3 SPACES}DIST.  ANG
        . W/ SUN{4 SPACES}R.
        A"
HD 125Ø PRINT "{38 U}":PRINT

OO 126Ø FOR X=1 TO 6:A2=Y1/P
        (X,2)-INT(Y1/P(X,2))
        :Q3=1
FM 127Ø A2=A2*36Ø+P(X,1):IF
        A2>36Ø THEN A2=A2-36
        Ø
PO 128Ø E=18Ø+A:IF E>36Ø THE
        N E=E-36Ø
LP 129Ø E1=ABS(E-A2):IF E1>1
        8Ø THEN E1=36Ø-E1
HD 13ØØ GOSUB 153Ø:E1=E1*O9:
        P5=P(X,3):IF X=3 THE
        N GOSUB 199Ø
FK 131Ø P(X,4)=SQR(1+P5^2-2*
        P5*COS(E1)):XX=(P5^2
        -1-P(X,4)^2)/(-2*P(X
        ,4))
```

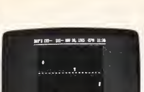
```
MI 1320 P(X,5)=-ATN(XX/SQR(-XX*XX+1))+PI/2:P(X,4)=INT(P(X,4)*ES+0.5):P(X,5)=P(X,5)/D9
LA 1330 ZZ=P(X,5):GOSUB 230:P(X,5)=ZZ:Q1$=STR$(P(X,4)):Q2$=STR$(P(X,5))
LC 1340 Q1=LEN(Q1$):Q2=LEN(Q2$):GOSUB 1630
DG 1350 PRINT P$((X-1)*7+1,X*7);:POKE 85,14-Q1:PRINT Q1$;:POKE 85,22-Q2:PRINT Q2$;:IF Q3=-1 THEN PRINT "@W";
AL 1360 IF Q3=1 THEN PRINT "@E";
LN 1370 GOSUB 1680:Q4$=STR$(Q4):Q5$=STR$(Q5):IF Q5<10 THEN ZZ$="0":ZZ$(2,LEN(Q5$)+1)=Q5$:Q5$=ZZ$
BL 1380 Q4$(LEN(Q4$)+1,LEN(Q4$)+1)=":":Q4$(LEN(Q4$)+1,LEN(Q4$)+LEN(Q5$))=Q5$:Z=LEN(Q4$)
AD 1390 PRINT ;:POKE 85,26:PRINT QQ$;:POKE 85,34-Z:PRINT Q4$:NEXT X:PRINT "{2 DOWN}* - VISIBLE AT 9:00 P.M."
JB 1400 PRINT "{2 DOWN}SUN'S R.A. -------";SPC$(1,Q8);A3$:PRINT "R.A. AT 9:00PM ---";SPC$(1,Q9);A5$
DM 1410 PRINT "{DOWN}-S- FOR DAYS SKY -D- FOR NEW DATE":GOTO 1930
BH 1420 A2=K1*A/360:IF A2>K1 THEN A2=A2-K1
MP 1430 A3=INT(A2/60):A4=A2-A3*60:A5=A3+9:IF A5>23 THEN A5=A5-24
MG 1440 A4=INT(A2-A3*60+0.5):IF A4=60 THEN A4=0:A3=A3+1
IA 1450 IF A3=24 THEN A3=0
HB 1460 AA=A3*60+A4:GOTO 1780
CB 1470 M1=(Y1/M9-INT(Y1/M9))*M9+10:IF M1>M9 THEN M1=M1-M9
PE 1480 GOSUB 2290:M8=360*M2:IF M8>180 THEN L$="W"
FJ 1490 IF M8<=180 THEN L$="E"
OM 1500 IF M8>180 THEN M8=360-M8
DD 1510 ZZ=M1:GOSUB 220:M1=ZZ:ZZ=M8:GOSUB 220:M8=ZZ:YY=INT(7*(Y1/7-INT(Y1/7))+0.2):IF YY=0 THEN YY=7
PG 1520 K$=J$(YY*3-2,YY*3):RETURN
MK 1530 Q3=0:Q1=E+180:IF Q1>360 THEN 1570
EO 1540 IF A2>E AND A2<Q1 THEN 1560
NH 1550 Q3=1:RETURN
AF 1560 Q3=-1:RETURN
OG 1570 Q1=Q1-360:IF A2<=360 AND A2>E THEN 1560
JK 1580 IF Q3<>0 THEN RETURN
HL 1590 IF A2>0 AND A2<=Q1 THEN 1560
JD 1600 IF Q3<>0 THEN RETURN
IE 1610 IF A2>Q1 THEN 1550
KJ 1620 RETURN
HK 1630 Q5=Q3*P(X,5)*4+AA:IF Q5<0 THEN Q5=Q5+K1
LL 1640 IF Q5>K1 THEN Q5=Q5-K1
OM 1650 P(X,6)=Q5:Q4=INT(Q5/60):Q5=INT(Q5-Q4*60+0.5):IF Q5=60 THEN Q5=0:Q4=Q4+1
KF 1660 IF Q4=24 THEN Q4=0
KO 1670 RETURN
OG 1680 SU=A5*60+A4:PS=SU+360:MS=SU-360:IF PS>K1 THEN PS=PS-K1
LO 1690 IF MS<0 THEN MS=MS+K1
MC 1700 IF MS>PS THEN 1730
JL 1710 IF P(X,6)<PS AND P(X,6)>MS THEN 1760
CL 1720 QQ$=" ":RETURN
GK 1730 IF P(X,6)<K1 AND P(X,6)>MS THEN 2000
JC 1740 IF P(X,6)<PS THEN 1760
MA 1750 GOTO 1720
FJ 1760 QQ$="*":RETURN
MA 1770 PRINT :PRINT K$;"--";H$;" ";D;",";Y;:POKE 85,20:PRINT ABS(LL);LL$;:RETURN
HK 1780 A3$=STR$(A3):A4$=STR$(A4)
NL 1790 IF A4<10 THEN ZZ$="0":ZZ$(2,2)=A4$:A4$=ZZ$
ML 1800 A3$(LEN(A3$)+1,LEN(A3$)+1)=":":A3$(LEN(A3$)+1,LEN(A3$)+LEN(A4$))=A4$:A5$=STR$(A5)
PD 1810 A5$(LEN(A5$)+1,LEN(A5$)+1)=":":A5$(LEN(A5$)+1,LEN(A5$)+LEN(A4$))=A4$
AG 1820 Q8=7-LEN(A3$):Q9=7-LEN(A5$):RETURN
NK 1830 LY=0:IF Y/4=INT(Y/4) THEN LY=1
AJ 1840 IF Y/100=INT(Y/100) AND Y/400<>INT(Y/400) THEN LY=0
IM 1850 IF Y/1000=INT(Y/1000) AND Y/4000=INT(Y/4000) THEN LY=0
KP 1860 RETURN
NA 1870 Y9=Y+1:IF Y9/4=INT(Y9/4) THEN ZY=1
PP 1880 IF Y9/100=INT(Y9/100) AND Y9/400<>INT(Y9/400) THEN ZY=0
ID 1890 IF Y9/1000=INT(Y9/1000) AND Y9/4000=INT(Y9/4000) THEN ZY=0
FB 1900 Y1=Y-1977:Y1=Y1*365+INT(Y1/4)+D1:IF Y<2000 THEN 1920
AH 1910 Y1=Y1-INT((Y-2001)/100)+INT((Y-2001)/400)-INT((Y-1)/4000)
KM 1920 RETURN
PG 1930 XX=VAL(STR$(0)):GET #1,I:Q$=CHR$(I)
GB 1940 IF Q$="D" THEN 920
HB 1950 IF (Q$="S" OR Q$="T") AND S1=1 THEN 370
JK 1960 IF Q$="P" THEN 1230
GD 1970 IF Q$="L" AND S1=1 THEN 2560
NI 1980 GOTO 1930
FP 1990 P5=1.37634408:K5=A2*4
FN 2000 K5=ABS(K5-1233.73)*90/K1:K5=K5*D9:K5=SIN(K5)*0.322581224:P5=P5+K5:RETURN
JB 2010 IF CC<=1 THEN CC=CC+84
MN 2020 CD$=CC$(CC-1,CC+41)
MP 2030 IF CD$(2,2)<>"■" AND CD$(3,3)="■" THEN CD$=CD$(1,40):GOTO 2060
JM 2040 IF CD$(41,41)<>"■" AND CD$(40,40)="■" THEN CD$=CD$(3,42):GOTO 2060
GO 2050 CD$=CD$(2,41)
KI 2060 RETURN
BI 2070 DATA 365.26,29.53059,59.818184,42.719626,262.364294,52.916763
JD 2080 DATA 134.69697,218.79464,87.97,224.7,686.98
FL 2090 DATA 4332.79813,10759.7195,30686.5884
HM 2100 DATA MERCURY,.3871,VENUS,.7233,MARS,1.5237,JUPITER,5.2028
IH 2110 DATA SATURN ,9.5308,URANUS,19.182
AH 2120 DATA 56,68,68,68,56,0,0,0,255,239,199,131,131,199,239,255
CJ 2130 DATA 195,129,153,153,195,231,129,231,252,249,195,153,153,153,199,255
AD 2140 DATA 255,195,189,129,129,189,195,255,252,193,145,137,153,131,63,127
OH 2150 DATA 255,153,153,153,219,231,255,255
ED 2160 DATA 245,234,213,202,213,139,7,31
KM 2170 DATA SA,SC,LI,VI,LE,CA,GE,TA,AR,PI,AQ,CP
KN 2180 DATA NEW,WAXING CRESCENT,1ST QUARTER,WAXING GIBBOUS,FULL
BF 2190 DATA WANING GIBBOUS,3RD QUARTER,WANING CRESCENT
FI 2200 DATA 1770,1719,1620,1500,1418,1365,1335,1310,1290,1275,1260
DA 2210 DATA 1238,1220,1200,1178,1115,915,720,660,640,625,610,255,255,0,0,0,0,0
BF 2220 POKE 756,PEEK(106)+1:PRINT "{CLEAR}{6 DOWN}{11 SPACES}**** SKYSCAPE ****"
IN 2230 PRINT "{2 DOWN}{10 SPACES}REDEFINING CHARACTERS"
EE 2240 CHBAS=(PEEK(106)+1)*256:FOR I=0 TO 1023:POKE CHBAS+I,PEEK(57344+I):NEXT I
KJ 2250 RETURN
MD 2260 PRINT "-N- TO RE-INPUT OR RETURN TO CONTINUE"
MK 2270 GET #1,ZZ:Z$=CHR$(ZZ)
KM 2280 RETURN
LH 2290 M2=M1/M9:IF M1<1 OR M1>28.5 THEN M3=1
MG 2300 IF M1>=1 AND M1<6.9 THEN M3=2
MP 2310 IF M1<=8 AND M1>6.9 THEN M3=3
LM 2320 IF M1>8 AND M1<14.2 THEN M3=4
MG 2330 IF M1>=14.2 AND M1<=15.2 THEN M3=5
FA 2340 IF M1>15.2 AND M1<21.6 THEN M3=6
MO 2350 IF M1>=21.6 AND M1<=22.6 THEN M3=7
```

```
JJ 2360 IF M1>22.6 AND M1<=2
        B.5 THEN M3=B
KM 2370 RETURN
HA 23B0 B$="":IF Y<>19B5 AND
         Y<>19B6 THEN 750
I6 2390 IF (Y=1985 AND D1<30
        5) DR (Y=1986 AND D1
        >149) THEN 750
EL 2400 HD=D1+365:IF HD>516
        THEN HD=HO-365
JB 2410 H1=(HD-295)/10:HD=IN
        T(H1):H1=H1-HD
NF 2420 T4=HC(HD)-HC(HD+1):T
        4=HC(HD)-H1*T4:IF T4
        >K1 THEN T4=T4-K1
DP 2430 GDSUB B00:IF Y9=999
        THEN 750
LD 2440 GDSU8 890:IF T4>1115
         AND T4<1200 THEN U9
        =U9+40
AF 2450 IF T4>1290 THEN U9=U
        9-40
HE 2460 IF T4>615 AND T4<=11
        15 THEN U9=U9+80
DE 2470 U(7)=U9:B$="{E}HALLE
        Y'S CDMET":GDTD 710
MC 24B0 GDSUB B90:PK=SCREEN+
        399-Y9+U9+LB
II 2490 IF LL<0 THEN PK=2*SC
        REEN+199+B0*XX-PK
KH 2500 RETURN
FP 2510 LL$="@N":IF LL<0 THE
        N LL$="@S"
LJ 2520 L1=ABS(LL):IF ABS(LL
        )<24 THEN L1=40
OP 2530 LC=INT((L1-40)/7+0.5
        ):LB=LC*40:D1=VAL(D$
        ((M*3)-2,M*3))+O
6C 2540 IF ABS(LL)<24 THEN L
        B=40*INT(ABS(LL)/7+0
        .5)
KM 2550 RETURN
6B 2560 PRINT "{CLEAR}{ODWN}
        {6 SPACES}***********
        * SKYSCAPE *********
        **":PRINT "{DDWN}LAT
        ITUDE CHANGE"
6D 2570 PRINT "{15 U}":GDSUB
         1770
AD 25B0 PRINT "{3 DOWN}":PRI
        NT "INPUT NEW LATITU
        DE";:INPUT LL:PRINT
        :PRINT
IF 2590 IF ABS(LL)>90 THEN P
        RINT DO$:GOTD 4560
BP 2600 GDSU8 2260:IF Z$="N"
         THEN 2510
MO 2610 GDSUB 2510:I$="S":GD
        TD 1950
OI 2620 FDR X=SCREEN+6B0 TD
        SCREEN+699:U1=PEEK(X
        ):U2=PEEK(2*SCREEN+1
        399-X):PDKE X,U2:PDK
        E 2*SCREEN+1399-X,U1
        :NEXT X
6J 2630 FOR X=SCREEN+680 TD
        SCREEN+71B:IF PEEK(X
        )=12B THEN 2650
EL 2640 U1=PEEK(X):U2=PEEK(X
        +1):PDKE X,U2:PDKE X
        +1,U1:X=X+1
H0 2650 NEXT X:RETURN
```

Program 3: IBM PC/PCjr Skyscape

Version by Tim Victor, Editorial Programmer

```
MP 100 KEY OFF:WIDTH 80:COLOR 0,
       0,0:CLS
GK 110 GOSUB 2210
FL 120 D$="000031059090120151181
```



"Skyscape" on the IBM PC/PCjr.

```
       212243273304334":K1=1440:
       DIM HC(22):MM$="041079040
       "
LI 130 M$="286317345011041072102
       133164194225255":D$(1)="S
       ":D$(2)="N":ES=93
DJ 140 A$="JANFEBMARAPRMAYJUNJUL
       AUGSEPOCTNOVDEC":OO$="DUT
        OF RANGE!":DG$=CHR$(248)
HH 150 MO$="312831303130313130313
       3031":O9=ATN(1)/45:READ E
       E:READ M9:DIM P(6,6)
MM 160 DEF FNR(X)=INT(X*10+.5)/1
       0
JA 170 DEF FNS(X)=INT(X*100+.5)/
       100
AA 180 FOR Y=1 TO 2:FOR X=1 TO 6
       :READ P(X,Y):NEXT:NEXT:Y=
       0
NM 190 FOR X=1 TO 6:READ P$(X),P
       (X,3):NEXT
IP 200 FOR X=1 TO 7:READ PP(X):N
       EXT
IK 210 J$="SATSUNMDNTUEWEDTHUFRI
       ":FOR X=1 TO 12:READ F$
FP 220 CC$=CC$+"     "+F$:NEXT:C
       C$=CC$+CC$:F$=RIGHT$(CC$,
       9):CC$=F$+CC$
IL 230 FOR X=1 TO B:READ PH$(X):
       NEXT
JH 240 FOR X=1 TO 22:READ HC(X):
       NEXT:GDTO 880
JE 250 CC=MT-720:IF CC<0 THEN CC
       =CC+K1
MI 260 CC=CC/120:CD=CC-INT(CC):C
       C=INT(CC):CD=INT(CD*7+.2)
       :CC=B1-(CC*7+CD)
JP 270 GOSUB 2060:IF LL<0 THEN G
       OSUB 2610
MM 280 PRINT CD$:RETURN
LC 290 LOCATE 24,20:PRINT SPC(40
       );
EH 300 LOCATE 4,SL:PRINT "** DAY
       S SKY **":LOCATE 5,SL:PRI
       NT "---------------"
LN 310 LOCATE 7,SL:PRINT "INPUT
       THE TIME:":LOCATE 8,SL:PR
       INT "---------------"
NH 320 LOCATE 9,SL:PRINT "HOUR (
       0-23)";:GOSUB 2450:IF I$<
       >"" THEN T1=VAL(I$)
HM 330 IF T1<0 OR T1>23 THEN LOC
       ATE 10,SL+3:PRINT OD$:GDT
       D 320
PC 340 LOCATE 11,SL:PRINT "MINUT
       E (0-59)";:GOSUB 2450:IF
       I$<>"" THEN T2=VAL(I$)
NK 350 IF T2<0 OR T2>59 THEN LOC
       ATE 12,SL:PRINT OD$:GDTO
       340
IA 360 R$=RIGHT$(STR$(T1),2):T$=
       RIGHT$(STR$(T2),2):IF T2<
       10 THEN T$="0"+RIGHT$(T$,
       1)
KP 370 LOCATE 14,SL:PRINT "TIME-
       - "R$":"T$
CN 380 LOCATE 24,20:GOSUB 2230:I
       F I$="N" THEN 290
KN 390 CDLOR 3,4:CLS:T3=T1*60+T2
       +AA-720:IF T3<0 THEN T3=T
       3+K1
LK 400 IF T3>K1 THEN T3=T3-K1
JL 410 MT=T3-360:IF MT<0 THEN MT
       =MT+K1
OD 420 PT=T3+360:IF PT>K1 THEN P
       T=PT-K1
HN 430 LOCATE 2,18:PRINT "OAY'S
       SKY--  ";:GDSUB 1800:PRIN
       T "  "R$":"T$
JG 440 LDCATE 3,18:PRINT"-------
       -----------------------
       ------------"
IH 450 CDLDR 7,1:TM=VAL(R$+"."+T
       $):IF TM<6 DR TM>18 THEN
       CDLOR 7,0
DB 460 XX=7+LC:FOR X=1 TO 14:LDC
       ATE 3+X,20:IF X=XX THEN 4
       80
EI 470 PRINT SPC(40);:GDTD 490
NB 4B0 PRINT " - - - - - - - - - -
        - - - - - - - - - - -  ";
I6 490 NEXT:LDCATE 18,20:CDLOR 0
       ,6:GDSUB 250:LOCATE 19,20
       :CDLOR 7,1:IF LL<0 THEN 5
       20
EJ 500 IF LL>24 THEN PRINT "E"SP
       C(18)"S"SPC(19)"W":GDTD 5
       40
6F 510 PRINT "UP-NORTH      ----D
       VERHEAD      DDWN-SDUTH":G
       DTD 540
BJ 520 IF LL<-24 THEN PRINT "E"S
       PC(18)"N"SPC(19)"W":GDTD
       540
SG 530 PRINT "UP-SOUTH      ----D
       VERHEAD      ODWN-NDRTH"
PE 540 T4=AA:GDSUB 780:Y8=B88
6M 550 IF Y9=999 THEN 590
PC 560 GDSUB 2460:Y8=Y9:IF A1<0
       THEN 590
EM 570 IF U9>17 OR U9<4 THEN 590
CL 580 COLOR 7,1:LOCATE U9,59-Y9
       :PRINT CHR$(42)
OB 590 T4=AA+M2*K1:IF T4>K1 THEN
        T4=T4-K1
EM 600 COLOR 7,1:IF TM<6 OR TM>1
       8 THEN CDLDR 7,0
HJ 610 GDSUB 780:IF Y9=999 THEN
       650
KF 620 MM=INT(M1/9.B3333)+1:GDSU
       B B60
MK 630 GOSUB 2460:IF U9>17 OR U9
       <4 THEN 650
OK 640 LOCATE U9,59-Y9:PRINT CHR
       $(MM):IF ABS(Y8-Y9)<=.5 T
       HEN CDLDR 1,7:LOCATE U9,5
       9-Y9:PRINT CHR$(79):COLDR
        7,1
AC 650 FOR X=1 TO 7:IF X=7 THEN
       2350
JN 660 T4=P(X,6):GOSU8 780:IF Y9
       =999 THEN 730
IK 670 U9=SIN((P(X,6)/4)/(1/O9))
       :U9=-3*U9+.5
HD 680 GOSU8 2470
KA 690 IF U9<4 OR U9>17 THEN 730
DB 700 Z=SCREEN(U9,59-Y9)
BL 710 IF Z<>32 ANO Z<>45 THEN U
       9=U9+SGN(LL)+(LL=0):GOTO
       700
DP 720 LOCATE U9,59-Y9:PRINT CHR
       $(PP(X));
OJ 730 NEXT
DE 740 LOCATE 21,14:COLOR 3,4:FO
       R X=1 TD 6:PRINT CHR$(PP(
       X));P$(X);"  ";:NEXT
JA 750 LOCATE 22,14:PRINT "*SUN
         )O(MDON     NEW MOON +
        SUN  ";8$
PC 760 LOCATE 22,33:COLOR 4,3:PR
       INT "O":COLDR 3,4
HM 770 LOCATE 24,20:PRINT "T- NE
```

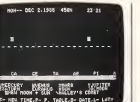
```
W TIME,P- P. TABLE,D- DAT
E,L- LAT";:SL=62:GDTD 198
Ø
MI 780 Y9=999:IF MT<PT THEN 820
NN 790 IF T4<MT AND T4>PT THEN R
ETURN
FK 800 IF T4<MT OR T4>PT THEN T4
=T4+K1
HM 810 GOTO 830
LB 820 IF T4<MT OR T4>PT THEN RE
TURN
FA 830 Y9=INT((T4-MT)/18+.5):IF
Y9=40 THEN Y9=39
MK 840 RETURN
FC 850 U9=SIN((T4/4)/(1/09)):U9=
INT(-3*U9+.5):RETURN
FL 860 MM=VAL(MID$(MM$,3*MM-2,3)
):IF L<Ø AND MM<>81 THEN
MM=ABS(MM-81)
NA 870 RETURN
DM 880 COLOR Ø,3:CLS:LOCATE 2,20
:PRINT "************** SK
YSCAPE **************":LD
CATE 4,10:PRINT "DATE INP
UT":S1=Ø
CA 890 LOCATE 5,10:PRINT "------
-----":IF Y<>Ø THEN LDCATE
4,40:GDSUB 1800
LI 900 LOCATE 7,4:PRINT "YEAR";:
GDSUB 2450:IF I$<>"" THEN
Y=VAL(I$)
MC 910 IF Y<1977 THEN PRINT "MUS
T BE AFTER 1977":GOTO 900
OJ 920 GOSUB 1880:LOCATE 9,4:PRI
NT "MDNTH (1-12)";:GDSUB
2450:IF I$<>"" THEN M=VAL
(I$)
MB 930 IF M<1 OR M>12 THEN PRINT
DD$:GOTO 920
MJ 940 OI=VAL(MID$(MO$,2*M-1,2))
:DI=DI-(M=2)*LY:DI$=STR$(
DI)
IO 950 LOCATE 11,4:PRINT"DAY (1-
"DI$")";:GDSUB 2450:IF I$
<>"" THEN D=VAL(I$)
FE 960 IF D<1 OR D>DI THEN PRINT
DD$:GOTO 950
LB 970 H$=MID$(A$,(M*3)-2,3):LOC
ATE 13,4:PRINT "LATTITUDE
(Ø-90)";:GDSUB 2450:IF I
$<>"" THEN LL=VAL(I$)
CH 980 GDSUB 2500
FB 990 IF ABS(LL)>90 THEN PRINT
DD$:GDTD 970
GP 1000 O1=VAL(MID$(D$,(M*3)-2,3
))+O:GOSUB 1920:IF M>2
THEN D1=D1+LY:Y1=Y1+LY
MA 1010 S=Ø:GOSUB 1540:LOCATE 4,
40:GDSUB 1800:LDCATE 5,4
Ø:PRINT "---------------
---------"
HN 1020 LDCATE 24,20:GOSUB 2230:
IF I$="N" THEN 880
BB 1030 LOCATE 24,20:PRINT SPC(4
Ø);
FM 1040 D2=VAL(MID$(M$,(M*3)-2,3
))+O:GOSUB 1920:IF M>
2 THEN D1=D1+LY:Y1=Y1+
LY
AD 1050 D3=D2-185:IF M=3 AND D<2
Ø THEN D2=D2+LY:D3=D3
+LY
OM 1060 IF D3<=Ø THEN A=180*D2/1
85:GOTO 1080
II 1070 A=(180*D3/(180+ZY))+180
LC 1080 IF A<180 THEN S=23.43333
*SIN(D9*D2*180/185)
LD 1090 IF A>180 THEN S=-23.4333
3*(SIN(D9*D3))
MD 1100 IF A>=360 THEN A=A-360
LC 1110 A=FNR(A)
KA 1120 S=FNR(S):A1=(SGN(LL)-(LL
=Ø))*S+90-ABS(LL):A1=FNR
(A1):GOSUB 1490:GOSUB 14
40
AF 1130 W=1-(LL<Ø):IF A1>90 THEN
A1=180-A1:W=3-W
PE 1140 LDCATE 7,36:PRINT "DAY O
F THE YEAR-----------
";D1
JD 1150 LDCATE 8,36:PRINT "SUN'S
GEDCENTRIC ANGLE----
";STR$(A);DG$
JM 1160 LOCATE 9,36:PRINT "SUN'S
DECLINATIDN---------
";STR$(S);DG$
DA 1170 LOCATE 10,36:PRINT "SUN'
S ALTITUDE AT NOON----
";STR$(A1);DG$;D$(W)
MM 1180 LDCATE 11,36:PRINT "SUN'
S RIGHT ASCENSIDN-----
";A3$
MF 1190 LOCATE 12,36:PRINT "R.A.
AT 9:ØØPM------------
";A5$
BC 1200 LDCATE 13,36:PRINT "MDON
'S AGE-----------------
";STR$(M1);"DY"
MM 1210 LDCATE 14,36:PRINT "MDDN
'S ELDNGATIDN---------
";STR$(M8);DG$;L$
GM 1220 LDCATE 15,36:PRINT "MOON
'S PHASE - "PH$(M3)
LD 1230 LDCATE 24,20:PRINT "-P-
PLANET TABLE, -D- NEW DA
TE";:GDTD 1980
JK 1240 CDLDR 7,5:CLS:LOCATE 2,2
Ø:PRINT "SKYSCAPE-   ";:
GDSUB 1800:S1=1
MF 1250 LOCATE 4,12:PRINT"** PLA
NET TABLE **":LOCATE 5,1
2:PRINT "---------------
---"
PD 1260 LDCATE 7,4:PRINT "PLANET
DIST.  ANG. W/ SUN
R.A."
MG 1270 LOCATE 8,4:PRINT "------
------------------------
------"
PO 1280 FOR X=1 TD 6:A2=Y1/P(X,2
)-INT(Y1/P(X,2)):Q3=1
GE 1290 A2=(A2*360)+P(X,1):IF A2
>360 THEN A2=A2-360
MK 1300 E=180+A:IF E>360 THEN E=
E-360
JL 1310 E1=ABS(E-A2):IF E1>180 T
HEN E1=360-E1
MP 1320 GOSUB 1560:E1=E1*O9:P5=P
(X,3):IF X=3 THEN GDSUB
2040
KI 1330 P(X,4)=SOR(1+P5^2-2*P5*C
DS(E1)):XX=((P5^2-1-P(X,
4)^2)/(-2*P(X,4)))
ME 1340 P(X,5)=-ATN(XX/SQR(-XX*X
X+1))+ATN(1)*2:P(X,4)=IN
T(P(X,4)*93+.5):P(X,5)=P
(X,5)/D9
JM 1350 P(X,5)=FNS(P(X,5)):Q1$=S
TR$(P(X,4)):Q2$=STR$(P(X
,5))
QI 1360 Q1=LEN(Q1$):O2=LEN(Q2$):
GOSUB 1660
FM 1370 LOCATE X+8,4:PRINT P$(X)
;TAB(18-Q1);Q1$;TAB(28-Q
2);Q2$;:IF Q3=-1 THEN PR
INT DG$"W";
BE 1380 IF Q3=1 THEN PRINT DG$"E
";
KM 1390 GOSUB 1710:Q4$=STR$(O4):
Q5$=STR$(Q5):IF Q5<10 TH
EN Q5$="Ø"+RIGHT$(O5$,1)
BM 1400 Q5$=RIGHT$(O5$,2):O4$=Q4
$+":"+Q5$:Z=LEN(O4$)
GC 1410 PRINT TAB(32);QO$;TAB(4Ø
-Z);Q4$:NEXT:LDCATE 15,4
:PRINT "* - VISIBLE AT 9
P.M."
JA 1420 LDCATE 17,4:PRINT "SUN'S
R.A. -------";SPC(Q8);A
3$:LDCATE 18,4:PRINT "R.
A. AT 9:ØØPM ---";SPC(Q9
);A5$
FQ 1430 SL=52:LDCATE 24,20:PRINT
"-S- FDR DAY'S SKY, -D-
FDR NEW OATE";:GOTO 198
Ø
FQ 1440 A2=K1*A/360:IF A2>K1 THE
N A2=A2-K1
NQ 1450 A3=INT(A2/6Ø):A4=A2-A3*6
Ø:A5=A3+9:IF A5>23 THEN
A5=A5-24
PM 1460 A4=INT(A2-A3*6Ø+.5):IF A
4=6Ø THEN A4=Ø:A3=A3+1
JO 1470 IF A3=24 THEN A3=Ø
GM 1480 AA=A3*6Ø+A4:GOTO 1840
EN 1490 M1=((Y1/M9)-INT(Y1/M9))*
M9+1Ø:IF M1>M9 THEN M1=M
1-M9
PK 1500 GOSUB 2260:M8=360*M2:IF
M8>180 THEN L$="W"
MB 1510 IF M8<=180 THEN L$="E"
KN 1520 IF M8>180 THEN M8=360-M8
HP 1530 M1=FNR(M1):M8=FNR(M8):RE
TURN
FL 1540 YY=INT(7*(Y1/7-INT(Y1/7)
)+.2):IF YY=Ø THEN YY=7
MH 1550 K$=MIO$(J$,(YY*3)-2,3):R
ETURN
CM 1560 Q3=Ø:O1=E+180:IF Q1>360
THEN 1600
IE 1570 IF A2>E AND A2<O1 THEN 1
590
DM 1580 Q3=1:RETURN
BA 1590 O3=-1:RETURN
LA 1600 O1=Q1-360:IF A2<=360 ANO
A2>E THEN 1590
MK 1610 IF Q3<>Ø THEN RETURN
EM 1620 IF A2>Ø ANO A2<=O1 THEN
1590
MA 1630 IF Q3<>Ø THEN RETURN
FD 1640 IF A2>O1 THEN 1580
JG 1650 RETURN
MK 1660 Q5=Q3*P(X,5)*4+AA:IF Q5<
Ø THEN Q5=Q5+K1
NN 1670 IF Q5>K1 THEN Q5=Q5-K1
EL 1680 P(X,6)=Q5:Q4=INT(O5/6Ø):
Q5=INT(Q5-Q4*6Ø+.5):IF Q
5=6Ø THEN Q5=Ø:Q4=Q4+1
IH 1690 IF Q4=24 THEN Q4=Ø
JJ 1700 RETURN
MM 1710 SU=A5*6Ø+A4:PS=SU+36Ø:MS
=SU-36Ø:IF PS>K1 THEN PS
=PS-K1
MG 1720 IF MS<Ø THEN MS=MS+K1
LE 1730 IF MS>PS THEN 1760
DG 1740 IF P(X,6)<PS AND P(X,6)>
MS THEN 1790
EI 1750 QQ$=" ":RETURN
GK 1760 IF P(X,6)<K1 ANO P(X,6)>
MS THEN 1790
LJ 1770 IF P(X,6)<PS THEN 1790
CB 1780 GOTD 1750
LC 1790 QQ$="*":RETURN
BH 1800 LL$=RIGHT$(STR$(ABS(LL))
,2):IF ABS(LL)<1Ø THEN L
L$=" "+RIGHT$(LL$,1)
MK 1810 PRINT K$;"-- ";H$;STR$(O
);",";Y;" ";LL$;DG$;:PRI
NT MIO$("SN",(LL<Ø)+2,1)
;
LN 1820 IF O<1Ø THEN PRINT " ";
JE 1830 RETURN
MB 1840 A4$=RIGHT$(STR$(A4),2)
NP 1850 IF A4<1Ø THEN A4$="Ø"+RI
GHT$(A4$,1)
MP 1860 A3$=STR$(A3)+":"+A4$:A5$
=STR$(A5)+":"+A4$
GJ 1870 Q8=7-LEN(A3$):Q9=7-LEN(A
5$):RETURN
MO 1880 LY=Ø:IF Y/4=INT(Y/4) THE
N LY=1
GI 1890 IF Y/1ØØ=INT(Y/1ØØ) AND
Y/4ØØ<>INT(Y/4ØØ)THEN LY
=Ø
FM 1900 IF Y/1ØØØ=INT(Y/1ØØØ) AN
O Y/4ØØØ=INT(Y/4ØØØ) THE
N LY=Ø
```

```
JA 1910 RETURN
MB 1920 Y9=Y+1:IF Y9/4=INT(Y9/4)
         THEN ZY=1
LO 1930 IF Y9/100=INT(Y9/100) AN
        D Y9/400<>INT(Y9/400) TH
        EN ZY=0
OI 1940 IF Y9/1000=INT(Y9/1000)
        AND Y9/4000=INT(Y9/4000)
         THEN ZY=0
BF 1950 Y1=Y-1977:Y1=Y1*365+INT(
        Y1/4)+D1:IF Y<2000 THEN
        1970
OC 1960 Y1=Y1-INT((Y-2001)/100)+
        INT((Y-2001)/400)-INT((Y
        -1)/4000)
KC 1970 RETURN
JB 1980 GOSUB 2240
JC 1990 IF I$="D" THEN 880
JH 2000 IF (I$="S" OR I$="T") AN
        D S1=1 THEN 290
NK 2010 IF I$="P" THEN 1240
GG 2020 IF I$="L" AND S1=1 THEN
        2540
EE 2030 GOTO 1980
AB 2040 P5=1.376344:K5=A2*4
LC 2050 K5=ABS(K5-1233.73)*90/K1
        :K5=K5*D9:K5=SIN(K5)*.32
        25812:P5=P5+K5:RETURN
FP 2060 IF CC<=0 THEN CC=CC+84
CD 2070 CD$=MID$(CC$,CC-1):IF MI
        D$(CD$,2,1)<>" " AND MID
        $(CD$,3,1)=" " THEN CD$=
        " "+CD$
GC 2080 IF MID$(CD$,40,1)=" " AN
        D MID$(CD$,41,1)<>"" THE
        N CD$=MID$(CD$,2)
JF 2090 CD$=MID$(CD$,2,40):RETUR
        N
HO 2100 DATA 356.26,29.53059,59.
        818184,42.719626,262.364
        394,52.9196763
OP 2110 DATA 134.69697.218.79464
        ,87.97,224.7,686.98
PO 2120 DATA 4332.79813,10759.71
        95,30686.5884
MA 2130 DATA "MERCURY",.3871,"VE
        NUS",.7233,"MARS",1.5237
        ,"JUPITER",5.2028
GK 2140 DATA "SATURN",9.5308,"UR
        ANUS",19.182
JL 2150 DATA 4,232,229,21,237,15
        7,231
GO 2160 DATA "SA","SC","LI","VI"
        ,"LE","CA","GE","TA","AR
        ","PI","AQ","CP"
OO 2170 DATA "NEW","WAXING CRESC
        ENT","1ST QUARTER","WAXI
        NG GIBBOUS","FULL"
HI 2180 DATA "WANING GIBBOUS","3
        RD QUARTER","WANING CRES
        CENT"
HM 2190 DATA 1770,1719,1620,1500
        ,1418,1365,1335,1310,129
        0,1275,1260
PO 2200 DATA 1238,1220,1200,1178
        ,1115,915,720,660,640,62
        5,610
LI 2210 CLS:LOCATE 7,12:PRINT "*
        *** SKYSCAPE ****"
IG 2220 RETURN
MH 2230 PRINT "-N- TO RE-INPUT O
        R RETURN TO CONTINUE";
EL 2240 I$="":WHILE LEN(I$)=0:I$
        =INKEY$:WEND:IF I$>"Z" T
        HEN I$=CHR$(ASC(I$)-32)
JP 2250 RETURN
MM 2260 M2=M1/M9:IF M1<1 OR M1>2
        8.5 THEN M3=1
EL 2270 IF M1>=1 AND M1<6.9 THEN
         M3=2
IH 2280 IF M1>=6.9 AND M1<=8 THE
        N M3=3
OA 2290 IF M1>8 AND M1<14.2 THEN
         M3=4
IC 2300 IF M1>=14.2 AND M1<15.2
        THEN M3=5
OK 2310 IF M1>=15.2 AND M1<21.6
        THEN M3=6
FC 2320 IF M1>=21.6 AND M1<=22.6
         THEN M3=7
OA 2330 IF M1>22.6 AND M1<=28.5
        THEN M3=8
JO 2340 RETURN
JL 2350 B$="":IF Y<>1985 AND Y<>
        1986 THEN 730
NO 2360 IF (Y=1985 AND D1<305) O
        R (Y=1986 AND D1>149) TH
        EN 730
OM 2370 HD=D1+365:IF HD>516 THEN
         HD=HD-365
DB 2380 H1=(HD-295)/10:HD=INT(H1
        ):H1=H1-HD
MA 2390 T4=HC(HD)-HC(HD+1):T4=HC
        (HD)-H1*T4:IF T4>1440 TH
        EN T4=T4-1440
MG 2400 GOSUB 780:IF Y9=999 THEN
         730
PK 2410 GOSUB 850:IF T4>1115 AND
         T4>1200 THEN U9=U9+1
HM 2420 IF T4>1290 THEN U9=U9-1
IP 2430 IF T4>615 AND T4<1115 TH
        EN U9=U9+2
PI 2440 U(7)=U9:B$=CHR$(PP(7))+"
        HALLEY'S COMET":GOTO 680
OM 2450 INPUT "";I$:RETURN
PJ 2460 GOSUB 850
BM 2470 IF LL>=0 THEN U9=LC+10+U
        9:GOTO 2490
CL 2480 U9=LC+10-U9:Y9=39-Y9
KP 2490 RETURN
IK 2500 LL$="@N":IF LL<0 THEN LL
        $="@S"
IO 2510 L1=ABS(LL):IF ABS(LL)<24
         THEN L1=40
KK 2520 LC=INT((L1-40)/7+.5):D1=
        VAL(MID$(D$,(M*3)-2,3))+
        D
JP 2530 RETURN
CP 2540 LOCATE 24,20:PRINT SPC(4
        0);
EH 2550 LOCATE 7,SL:PRINT "NEW L
        ATTITUDE":LOCATE 8,SL:PR
        INT "--------------"
DO 2560 LOCATE 9,SL:PRINT "LAT (
        0-90)";:GOSUB 2450:IF I$
        <>"" THEN LL=VAL(I$)
BJ 2570 IF ABS(LL)>90 THEN LOCAT
        E 10,SL+3:PRINT DD$:GOTO
         2560
LO 2580 LOCATE 24,20:GOSUB 2230:
        IF I$="N" THEN 2540
IE 2590 LOCATE 9,SL:PRINT SPC(80
        -SL);
GK 2600 GOSUB 2500:I$="S":GOTO 2
        000
JI 2610 CI=1:C2$=""
OB 2620 C1$=MID$(CD$,CI,1):IF C1
        $<>" " THEN 2640
FA 2630 C2$=C1$+C2$:CI=CI+1:GOTO
         2650
MN 2640 C2$=MID$(CD$,CI,2)+C2$:C
        I=CI+2
IC 2650 IF CI<41 THEN 2620
LO 2660 CD$=C2$:RETURN
```

Program 4: Apple Skyscape

Version by Tim Victor, Editorial Programmer

```
1D 60 GOSUB 1940
4B 70 D$ = "000031059090120151181
      212243273304334":K1 = 144
      0: DIM HC(22):MM$ = "0410B
      1040"
B4 80 M$ = "28631734501104107210
      213316419422525 5":D$(1) =
      "S":D$(2) = "N":ES = 93
23 90 A$ = "JANFEBMARAPRMAYJUNJU
      LAUGSEPOCTNOVDEC":DD$ = "O
```



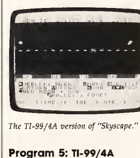
"Skyscape" on an Apple II-series computer.

```
      UT OF RANGE!!"
2A 100 MD$ = "312831303130313130
       313031":D9 = ATN (1) / 45
       : READ EE: READ M9: DIM P
       (6,6)
CB 110 DEF FN R(X) = INT (X * 10
       0 + .5) / 100
46 120 DEF FN S(X) = INT (X * 10
        + .5) / 10
B8 130 FOR Y = 1 TO 2: FOR X = 1
        TO 6: READ P(X,Y): NEXT
       : NEXT :Y = 0
73 140 FOR X = 1 TO 6: READ P$(X
       ),P(X,3): NEXT
14 150 FOR X = 1 TO 7:PP(X) = X
       + 85: NEXT
1F 160 J$ = "SATSUNMONTUEWEDTHUF
       RI": FOR X = 1 TO 12: REA
       D F$
B8 170 CC$ = CC$ + "      " + F$:
        NEXT :CC$ = CC$ + CC$:F$
        = RIGHT$ (CC$,9):CC$ = F
       $ + CC$
C1 180 FOR X = 1 TO 8: READ PH$(
       X): NEXT
2E 190 FOR X = 1 TO 22: READ HC(
       X): NEXT :R$ = "0":T$ = "
       00": GOTO 720
3B 200 CC = MT - 720: IF CC < 0
       THEN CC = CC + K1
2C 210 CC = CC / 120:CD = CC - I
       NT (CC):CC = INT (CC):CD
       = INT (CD * 7 + .2):CC =
       81 - (CC * 7 + CD)
51 220 GOSUB 1770: IF LL < 0 THE
       N GOSUB 5000
92 225 VTAB 17: PRINT CD$;: RETU
       RN
DB 230 HOME : HTAB 10: PRINT "**
        DAYS SKY **": VTAB 3: GO
       SUB 1550: HTAB 31: PRINT
       R$":"T$
DF 240 VTAB 5: HTAB 1: PRINT "IN
       PUT THE TIME:": PRINT "--
       --------------"
B8 245 PRINT : PRINT "     HOUR
       (0-23) ";: GOSUB 2240: IF
        I$ < > "" THEN T1 = VAL
       (I$)
B3 250 IF T1 < 0 OR T1 > 23 THEN
        PRINT DD$: GOTO 245
A3 255 PRINT : PRINT "   MINUTE
       (0-59) ";: GOSUB 2240: IF
        I$ < > "" THEN T2 = VAL
       (I$)
90 260 IF T2 < 0 OR T2 > 59 THEN
        PRINT DD$: GOTO 255
85 270 R$ = STR$ (T1):T$ = STR$
       (T2): IF LEN (T$) = 1 THE
       N T$ = "0" + T$
E8 280 VTAB 13: PRINT "TIME-- "R
       $":"T$
AA 290 PRINT : GOSUB 2020: IF I$
        = "N" THEN 230
1B 300 HOME :T3 = T1 * 60 + T2 +
        AA - 720: IF T3 < 0 THEN
        T3 = T3 + K1
```

```
2# 310 IF T3 > K1 THEN T3 = T3 -
       K1
B7 320 MT = T3 - 36Ø: IF MT < Ø
       THEN MT = MT + K1
FF 330 PT = T3 + 36Ø: IF PT > K1
       THEN PT = PT - K1
16 340 HTA8 4: GOSU8 155Ø: HTA8
       31: PRINT R$":"T$
F7 350 TM = VAL (R$ + "." + T$):
       IF TM > = 6 AND TM < = 1
       8 THEN INVERSE
B4 360 XX = 7 + LC: VTA8 3: HTA8
       1: FOR X = 1 TO 14: IF X
       = XX THEN GOTO 38Ø
CE 370 PRINT SPC( 4Ø);: GOTO 39Ø
#6 380 PRINT " - - - - - - - - - - -
       - - - - - - - - - - - -  ";
D2 390 NEXT X: NORMAL : GOSU8 2Ø
       Ø: INVERSE : IF LL < Ø TH
       EN 395
31 393 IF LL > 24 THEN PRINT "E"
       SPC( 18)"S" SPC( 19)"W":
       GOTO 4ØØ
3# 394 PRINT "UP-NORTH" SPC( 5)"
       ----OVERHEAO" SPC( 5)"OOW
       N-SOUTH": GOTO 4ØØ
D5 395 IF LL < - 24 THEN PRINT "
       W" SPC( 1B)"N" SPC( 19)"E
       ": GOTO 4ØØ
C# 397 PRINT "UP-SOUTH" SPC( 5)"
       ----OVERHEAO" SPC( 5)"OOW
       N-NORTH"
B9 400 T4 = AA: GOSUB 61Ø:YB = 8
       BB
43 410 IF Y9 = 999 THEN 45Ø
A6 420 GOSUB 4ØØØ:Y8 = Y9: IF A1
       < Ø THEN 45Ø
D6 430 IF U9 > 16 OR U9 < 3 THEN
       45Ø
## 440 VTAB U9: HTAB 4Ø - Y9: PR
       INT CHR$ (42)
97 450 T4 = AA + M2 * K1: IF T4
       > K1 THEN T4 = T4 - K1
E6 460 GOSU8 61Ø: IF Y9 = 999 TH
       EN 5ØØ
97 470 MM = INT (M1 / 9.83333) +
       1: GOSU8 71Ø
1D 480 GOSU8 4ØØØ: IF U9 > 16 OR
       U9 < 3 THEN 5ØØ
6B 490 VTA8 U9: HTA8 4Ø - Y9: PR
       INT CHR$ (MM);: IF ABS (Y
       8 - Y9) < = .5 THEN NORMA
       L : HTAB 4Ø - Y9: PRINT C
       HR$ (B1);: INVERSE
7B 500 FOR X = 1 TO 7: IF X = 7
       THEN 214Ø
6B 510 T4 = P(X,6): GOSU8 61Ø: I
       F Y9 = 999 THEN 56Ø
B7 520 U9 = SIN ((P(X,6) / 4) /
       (1 / 09)):U9 = INT ( - 3
       * U9 + .5)
13 530 GOSU8 4ØØ5: IF U9 < 3 OR
       U9 > 16 THEN 56Ø
BE 540 SR = INT ((U9 - 1) / 8):Z
       = PEEK (1Ø24 - SR * 984
       + (U9 - 1) * 12B + 39 - Y
       9): IF Z > 127 THEN Z = Z
       - 128
A# 545 IF Z < > 32 ANO Z < > 45
       THEN U9 = U9 + 2 * (LL >
       = Ø) - 1: GOTO 54Ø
2A 550 VTA8 U9: HTA8 4Ø - Y9: PR
       INT CHR$ (PP(X));
EB 560 NEXT X: NORMAL
F3 570 VTA8 2Ø: HTA8 1: PRINT "V
       MERCURY   WVENUS   XMARS
          YJUPITER"
2F 580 PRINT "ZSATURN    [URANUS
         *SUN       )Q(MOON"
99 590 HTA8 3: INVERSE : PRINT "
       Q";: NORMAL : PRINT "NEW
       MOON + SUN  "8$
92 600 PRINT : PRINT "T- NEW TIM
       E,P- P. TA8LE,O- OATE,L-
       LAT";: GOTO 17ØØ
7F 610 Y9 = 999: IF MT < PT THEN
       66Ø
36 620 IF (T4 > = MT) OR (T4 < =
       PT) THEN 64Ø
1C 630 RETURN
87 640 IF (T4 > = MT) ANO (T4 <
       = K1) THEN 6BØ
7C 650 T4 = T4 + K1: GOTO 68Ø
C4 660 IF (T4 > = MT) ANO (T4 <
       = PT) THEN GOTO 68Ø
24 670 RETURN
7A 680 Y9 = INT ((T4 - MT) / 18
       + .5): IF Y9 = 4Ø THEN Y9
       = 39
28 690 RETURN
6A 700 U9 = SIN ((T4 / 4) / (1 /
       09)):U9 = INT ( - 3 * U9
       + .5): RETURN
56 710 MM = VAL ( MID$ (MM$,3 *
       MM - 2,3)): IF LL < Ø AND
       MM < > B1 THEN MM = A8S
       (MM - 81)
2D 715 RETURN
DA 720 HOME : VTA8 2: HTA8 7: PR
       INT "*********** SKYSCAPE
       ***********": VTA8 4: PRIN
       T "OATE INPUT"
56 730 PRINT "-----------": IF Y
       < > Ø THEN VTA8 6: GOSU8
       155Ø: PRINT : PRINT
E5 740 PRINT "YEAR  ";: GOSU8 22
       4Ø: IF I$ < > "" THEN Y =
       VAL (I$)
14 745 IF Y < 1977 THEN PRINT "M
       UST BE AFTER 1977": GOTO
       74Ø
D3 750 GOSU8 16ØØ: PRINT : PRINT
       "MONTH (1-12) ";: GOSU8
       224Ø: IF I$ < > "" THEN M
       = VAL (I$)
B5 755 IF M < 1 OR M > 12 THEN P
       RINT OO$: GOTO 75Ø
65 760 OI = VAL ( MIO$ (MO$,2 *
       M - 1,2)):OI = OI + (M =
       2) * LY:OI$ = STR$ (OI):O
       I$ = RIGHT$ (OI$,2)
3B 770 PRINT : PRINT "OAY (1-"OI
       $") ";: GOSU8 224Ø: IF I$
       < > "" THEN O = VAL (I$)
BA 775 IF D < 1 OR O > OI THEN P
       RINT OO$: GOTO 77Ø
F2 7B0 H$ = MIO$ (A$,(M * 3) - 2
       ,3) + " ": PRINT : PRINT
       "LATITUOE (Ø-9Ø)";: GOSU8
       224Ø: IF I$ < > "" THEN
       LL = VAL (I$)
F8 786 GOSU8 45ØØ
E9 790 IF A8S (LL) > 9Ø THEN PRI
       NT OO$: GOTO 7BØ
68 800 PRINT : HTAB 5: GOSU8 129
       5: GOSUB 155Ø: PRINT : PR
       INT : GOSU8 2Ø2Ø: IF I$ =
       "N" THEN 72Ø
BD 820 O2 = VAL ( MIO$ (M$,(M *
       3) - 2,3)) + O: GOSU8 164
       Ø: IF M > 2 THEN O1 = O1
       + LY:Y1 = Y1 + LY
2D 830 O3 = O2 - 185: IF M = 3 A
       NO O < 2Ø THEN O2 = O2 +
       LY:O3 = O3 + LY
F# 840 S = Ø: IF O3 < = Ø THEN A
       = 1BØ * O2 / 185: GOTO 8
       6Ø
E2 850 A = 18Ø * O3 / (18Ø + ZY)
       + 18Ø
B2 B60 IF A < > 1BØ THEN S = 23.
       43333333 * ( SIN (O9 * O2
       * 18Ø / 185))
D4 870 IF A > 18Ø THEN S = - 23.
       43333333 * ( SIN (O9 * O3
       ))
E9 880 IF A > = 36Ø THEN A = A -
       36Ø
B3 885 A = FN R(A)
E1 890 S = FN R(S):A1 = ( SGN (L
       L) + (LL = Ø)) * S + 9Ø -
       ABS (LL):A1 = FN R(A1):
       GOSUB 125Ø: GOSUB 12ØØ
67 B95 W = 2 - (LL < Ø): IF A1 >
       9Ø THEN A1 = 18Ø - A1:W
       = 3 - W
25 900 HOME : VTA8 2: GOSUB 155Ø
       : PRINT : PRINT "--------
       ----------------"
64 910 PRINT : PRINT "OAY OF THE
       YEAR-----------    ";O1
12 920 PRINT "SUNS GEOCENTRIC AN
       GLE-----      ";A;"@"
8D 930 PRINT "SUNS OECLINATION--
       -------      ";S;"@"
39 940 PRINT "SUNS ALTITUOE AT N
       OON-----      ";A1;"@";O$(W
       )
48 950 PRINT "SUNS RIGHT ASCENSI
       ON------      ";A3$
55 960 PRINT "R.A. AT 9:ØØPM----
       -------      ";A5$
8E 970 PRINT "MOONS AGE----------
       --------      ";M1;"OY"
FB 980 PRINT "MOONS ELONGATION--
       -------      ";M8;"@";L$
BE 990 PRINT "MOONS PHASE - "PH$
       (M3)
63 1000 VTAB 17: PRINT "-P- PLAN
       ET TABLE ,-O- NEW OATE":
       GOTO 17ØØ
F3 1010 HOME : HTA8 11: PRINT "*
       * PLANET TABLE **": VTA8
       3: GOSUB 155Ø:S1 = 1
CE 1020 VTA8 5: HTA8 1: PRINT "P
       LANET   OIST.  ANG. W/ S
       UN     R.A"
AA 1030 VTA8 6: PRINT "---------
       ---------------------------
       -----"
D4 1040 FOR X = 1 TO 6:A2 = Y1 /
       P(X,2) - INT (Y1 / P(X,
       2)):Q3 = 1
19 1050 A2 = (A2 * 36Ø) + P(X,1)
       : IF A2 > 36Ø THEN A2 =
       A2 - 36Ø
B1 1060 E = 18Ø + A: IF E > 36Ø
       THEN E = E - 36Ø
FD 1070 E1 = ABS (E - A2): IF E1
       > 1BØ THEN E1 = 36Ø - E
       1
24 1080 GOSUB 131Ø:E1 = E1 * O9:
       P5 = P(X,3): IF X = 3 TH
       EN GOSUB 175Ø
1D 1090 P(X,4) = SOR (1 + P5 ^ 2
       - 2 * 1 * P5 * COS (E1)
       ):XX = ((P5 ^ 2 - 1 - P(
       X,4) ^ 2) / ( - 2 * P(X,
       4)))
7D 1100 P(X,5) = - ATN (XX / SQR
       ( - XX * XX + 1)) + ATN
       (1) * 2:P(X,4) = INT (P
       (X,4) * 93 + .5):P(X,5)
       = P(X,5) / O9
56 1110 P(X,5) = FN S(P(X,5)):Q1
       $ = STR$ (P(X,4)):Q2$ =
       STR$ (P(X,5))
DB 1120 Q1 = LEN (Q1$):Q2 = LEN
       (Q2$): GOSUB 141Ø
#B 1130 PRINT P$(X); TAB( 14 - Q
       1);Q1$; TAB( 24 - Q2);Q2
       $;: IF Q3 = - 1 THEN PRI
       NT "@W";
DA 1140 IF Q3 = 1 THEN PRINT "@E
       ";
78 1150 GOSUB 146Ø:Q4$ = STR$ (Q
       4):Q5$ = STR$ (Q5): IF Q
       5 < 1Ø THEN Q5$ = "Ø" +
       RIGHT$ (Q5$,1)
9D 1160 Q5$ = RIGHT$ (Q5$,2):Q4$
       = Q4$ + ":" + Q5$:Z = L
       EN (Q4$)
71 1170 PRINT TAB( 28)QQ$ TAB( 3
       6 - Z)Q4$: NEXT : VTAB 1
       4: PRINT "* - VISIBLE AT
       9 P.M."
```

```
5F 1180 VTAB 17: PRINT "SUNS R.A
     . --------" SPC( Q8)A3$:
     PRINT "R.A. AT 9:00PM -
     --" SPC( Q9)A5$
15 1190 VTAB 21: PRINT "-S- FDR
     DAYS SKY -D- FDR NEW DAT
     E": GDTD 1700
D4 1200 A2 = K1 * A2 / 360: IF A2
     > K1 THEN A2 = A2 - K1
77 1210 A3 = INT (A2 / 60):A4 =
     A2 - A3 * 60:A5 = A3 + 9
     : IF A5 > 23 THEN A5 = A
     5 - 24
27 1220 A4 = INT (A2 - A3 * 60 +
     .5): IF A4 = 60 THEN A4
     = 0:A3 = A3 + 1
93 1230 IF A3 = 24 THEN A3 = 0
8C 1240 AA = A3 * 60 + A4: GDTD
     1560
8D 1250 M1 = ((Y1 / M9) - INT (Y
     1 / M9)) * M9 + 10: IF M
     1 > M9 THEN M1 = M1 - M9
BC 1260 GDSUB 2050:M8 = 360 * M2
     : IF M8 > 180 THEN L$ =
     "W"
B4 1270 IF M8 < = 180 THEN L$ =
     "E"
EE 1280 IF M8 > 180 THEN M8 = 36
     0 - M8
56 1290 M1 = FN R(M1):M8 = FN R(
     M8)
88 1295 YY = INT (7 * (Y1 / 7 -
     INT (Y1 / 7)) + .2): IF
     YY = 0 THEN YY = 7
48 1300 K$ = MID$ (J$,(YY * 3) -
     2,3): RETURN
14 1310 Q3 = 0:Q1 = E + 180: IF
     Q1 > 360 THEN 1350
61 1320 IF A2 > E AND A2 < Q1 TH
     EN 1340
D2 1330 Q3 = 1: RETURN
18 1340 Q3 = - 1: RETURN
6E 1350 Q1 = Q1 - 360: IF A2 < =
     360 AND A2 > E THEN 134
     0
6D 1360 IF Q3 < > 0 THEN RETURN
6A 1370 IF A2 > 0 AND A2 < = Q1
     THEN 1340
95 1380 IF Q3 < > 0 THEN RETURN
44 1390 IF A2 > Q1 THEN 1330
19 1400 RETURN
FB 1410 Q5 = Q3 * P(X,5) * 4 + A
     A: IF Q5 < 0 THEN Q5 = Q
     5 + K1
8C 1420 IF Q5 > K1 THEN Q5 = Q5
     - K1
92 1430 P(X,6) = Q5:Q4 = INT (Q5
     / 60):Q5 = INT (Q5 - Q4
     * 60 + .5): IF Q5 = 60
     THEN Q5 = 0:Q4 = Q4 + 1
29 1440 IF Q4 = 24 THEN Q4 = 0
ED 1450 RETURN
D8 1460 SU = A5 * 60 + A4:PS = S
     U + 360:MS = SU - 360: I
     F PS > K1 THEN PS = PS -
     K1
96 1470 IF MS < 0 THEN MS = MS +
     K1
5A 1480 IF MS > PS THEN 1510
26 1490 IF P(X,6) < PS AND P(X,6
     ) > MS THEN 1540
D2 1500 QQ$ = " ": RETURN
DE 1510 IF P(X,6) < K1 AND P(X,6
     ) > MS THEN 1540
46 1520 IF P(X,6) < PS THEN 1540
6A 1530 GDTD 1500
A3 1540 QQ$ = "*": RETURN
89 1550 PRINT K$"-- "H$;D","Y;"
     "$: IF LL < 10 THEN PRI
     NT " ";
AE 1555 PRINT ABS (LL);LL$;: RET
     URN
EF 1560 A3$ = STR$ (A3):A3$ = RI
     GHT$ (A3$,2):A4$ = STR$
     (A4):A4$ = RIGHT$ (A4$,2
     )
42 1570 IF A4 < 10 THEN A4$ = "0
     " + RIGHT$ (A4$,1)
3D 1580 A3$ = A3$ + ":" + RIGHT$
      (A4$,2):A5$ = STR$ (A5)
     :A5$ = RIGHT$ (A5$,2) +
     ":" + A4$
EB 1590 Q8 = 7 - LEN (A3$):Q9 =
     7 - LEN (A5$): RETURN
5D 1600 LY = 0: IF Y / 4 = INT (
     Y / 4) THEN LY = 1
49 1610 IF Y / 100 = INT (Y / 10
     0) AND Y / 400 < > INT (
     Y / 400) THEN LY = 0
CF 1620 IF Y / 1000 = INT (Y / 1
     000) AND Y / 4000 = INT
      (Y / 4000) THEN LY = 0
E9 1630 RETURN
4B 1640 Y9 = Y + 1: IF Y9 / 4 =
     INT (Y9 / 4) THEN ZY = 1
8B 1650 IF Y9 / 100 = INT (Y9 /
     100) AND Y9 / 400 < > IN
     T (Y9 / 400) THEN ZY = 0
C8 1660 IF Y9 / 1000 = INT (Y9 /
      1000) AND Y9 / 4000 = I
     NT (Y9 / 4000) THEN ZY =
      0
B8 1670 Y1 = Y - 1977:Y1 = Y1 *
     365 + INT (Y1 / 4) + D1:
      IF Y < 2000 THEN 1690
DC 1680 Y1 = Y1 - INT ((Y - 2001
     ) / 100) + INT ((Y - 200
     1) / 400) - INT ((Y - 1)
      / 4000)
02 1690 RETURN
5D 1700 GET I$
F1 1710 IF I$ = "D" THEN 720
F9 1720 IF (I$ = "S" DR I$ = "T"
     ) AND S1 = 1 THEN 230
A9 1730 IF I$ = "P" THEN 1010
06 1735 IF I$ = "L" AND S1 = 1 T
     HEN 4550
76 1740 GDTD 1700
EB 1750 P5 = 1.376344086:K5 = A2
      * 4
97 1760 K5 = ABS (K5 - 1233.73)
     * 90 / K1:K5 = K5 * D9:K
     5 = SIN (K5) * .32258122
     4:P5 = P5 + K5: RETURN
B5 1770 IF CC < = 0 THEN CC = CC
      + 84
47 1780 CD$ = MID$ (CC$,CC - 1)
0F 1785 IF MID$ (CD$,2,1) < > "
     " AND MID$ (CD$,3,1) = "
      " THEN CD$ = " " + CD$
3D 1786 IF MID$ (CD$,41,1) = " "
      AND MID$ (CD$,42,1) < >
      " " THEN CD$ = MID$ (CD
     $,2)
D3 1788 CD$ = MID$ (CD$,2,40): R
     ETURN
8D 1790 DATA 365.26,29.53059,59.
     818184,42.719626,262.364
     4,52.916763
91 1800 DATA 134.69697,218.79464
     ,87.97,224.7,686.98
39 1810 DATA 4332.79813,10759.71
     95,30686.5884
25 1820 DATA "MERCURY",.3871,"VE
     NUS",.7233,"MARS",1.5237
     ,"JUPITER",5.2028
06 1830 DATA "SATURN",9.5308,"UR
     ANUS",19.182
A5 1890 DATA "SA","SC","LI","VI"
     ,"LE","CA","GE","TA","AR
     ","PI","AQ","CP"
15 1900 DATA "NEW","WAXING CRESC
     ENT","1ST QUARTER","WAXI
     NG GIBBDUS","FULL"
E4 1910 DATA "WANING GIBBDUS","3
     RD QUARTER","WANING CRES
     CENT"
05 1920 DATA 1770,1719,1620,1500
     ,1418,1365,1335,1310,129
     0,1275,1260
4A 1930 DATA 1238,1220,1200,1178
     ,1115,915,720,660,640,625
     ,610
D4 1940 PRINT CHR$ (17): HDME :
     VTAB 7: HTAB 12: PRINT "
     **** SKYSCAPE ****"
F7 1950 RETURN
AC 2020 PRINT "-N- TD RE-INPUT D
     R RETURN TD CDNTINUE"
6F 2030 GET I$: RETURN
36 2050 M2 = M1 / M9: IF M1 < 1
     DR M1 > 28.5 THEN M3 = 1
CA 2060 IF M1 > = 1 AND M1 < 6.9
      THEN M3 = 2
36 2070 IF M1 < = 8.0 AND M1 > =
      6.9 THEN M3 = 3
D9 2080 IF M1 > 8.0 AND M1 < 14.
     2 THEN M3 = 4
8B 2090 IF M1 > = 14.2 AND M1 <
     = 15.2 THEN M3 = 5
69 2100 IF M1 > 15.2 AND M1 < 21
     .6 THEN M3 = 6
6F 2110 IF M1 > = 21.6 AND M1 <
     = 22.6 THEN M3 = 7
34 2120 IF M1 > 22.6 AND M1 < =
     28.5 THEN M3 = 8
E0 2130 RETURN
12 2140 B$ = "": IF Y < > 1985 A
     ND Y < > 1986 THEN 560
41 2150 IF (Y = 1985 AND D1 < 30
     5) DR (Y = 1986 AND D1 >
     149) THEN 560
AB 2160 HD = D1 + 365: IF HD > 5
     16 THEN HD = HD - 365
D2 2170 H1 = (HD - 295) / 10:HD
     = INT (H1):H1 = H1 - HD
FA 2180 T4 = HC(HD) - HC(HD + 1)
     :T4 = HC(HD) - H1 * T4:
     IF T4 > 1440 THEN T4 = T
     4 - 1440
AB 2190 GDSUB 610: IF Y9 = 999 T
     HEN 560
3A 2200 GDSUB 700: IF T4 > 1115
     AND T4 < 1200 THEN U9 =
     U9 + 1
87 2210 IF T4 > 1290 THEN U9 = U
     9 - 1
A3 2220 IF T4 > 615 AND T4 < 111
     5 THEN U9 = U9 + 2
6E 2230 U(7) = U9:B$ = CHR$ (PP(
     7)) + "HALLEY'S CDMET":
     GDTD 530
4D 2240 INPUT I$: RETURN
BB 2250 VTAB 17: PRINT CD$;: RET
     URN
2F 4000 GDSUB 700
27 4005 IF LL > = 0 THEN U9 = LC
     + 9 + U9: GDTD 4008
EE 4006 U9 = LC + 9 - U9:Y9 = 39
     - Y9
15 4008 RETURN
AB 4500 LL$ = "@N": IF LL < 0 TH
     EN LL$ = "@S"
F3 4510 L1 = ABS (LL): IF L1 < 2
     4 THEN L1 = 40
44 4515 LC = INT ((L1 - 40) / 7
     + .5):D1 = VAL ( MID$ (D
     $,(M * 3) - 2,3)) + D
EA 4530 RETURN
53 4550 HDME : VTAB 2: HTAB 7: P
     RINT "********** SKYSCAP
     E **********": VTAB 4: P
     RINT "LATTITUDE CHANGE"
E9 4555 PRINT "----------------"
B3 4560 VTAB 8: PRINT "ENTER NEW
     LATTITUDE";: GDSUB 2240
     : IF I$ < > "" THEN LL =
     VAL (I$)
C9 4565 IF ABS (LL) > 90 THEN PR
     INT DD$;: GDTD 4560
E2 4570 GDSUB 2020: IF I$ = "N"
     THEN 4550
A8 4580 GDSUB 4500:I$ = "S": GDT
     D 1720
28 5000 CI = 1:C2$ = ""
25 5010 C1$ = MID$ (CD$,CI,1): I
     F C1$ < > " " THEN 5030
AA 5020 C2$ = C1$ + C2$:CI = CI
```

```
     + 1: GOTO 5Ø4Ø
85 5Ø3Ø C1$ = MID$ (CO$,CI,2):C2
     $ = C1$ + C2$:CI = C1 +
     2
F8 5Ø4Ø IF CI < 41 THEN 5Ø1Ø
58 5Ø5Ø CO$ = C2$: RETURN
```



The TI-99/4A version of "Skyscape."

Program 5: TI-99/4A Skyscape

Version by Patrick Parrish, Programming Supervisor

```
100 GOTO 13Ø
11Ø PK=PK-1Ø23 :: PKROW=I
    NT(PK/4Ø)-1 :: PKCOL=
    PK-(PKROW+1)*4Ø :: RE
    TURN
12Ø FOR I=1 TO LEN(QQ$)::
     CALL HCHAR(ROW,COL+I
    ,ASC(SEG$(QQ$,1,1))):
    : NEXT I :: RETURN
13Ø MM$="Ø9B1Ø8Ø99" :: CA
    LL CLEAR :: CALL SCRE
    EN(15):: DISPLAY AT(1
    1,6):"**** SKYSCAPE *
    ***" :: DISPLAY AT(22
    ,8):"INITIALIZING..."
14Ø D$="ØØØ31Ø59Ø9Ø12Ø15
    11812122432733Ø4334"
    :: K1=144Ø :: DIM HC(
    22):: M$="2B631734501
    1Ø41Ø721Ø2133164194 22
    5255"
15Ø ES=93 :: D1$(1)="S" :
    : D1$(2)="N"
16Ø A$="JANFEBMARAPRMAYJU
    NJULAUGSEPOCTNOVDEC"
    :: DD$="OUT OF RANGE!
    !" :: MD$="3128313Ø31
    3Ø31313Ø313Ø31" :: D9
    =PI/18Ø :: READ EE,M9
17Ø DIM P(6,6):: DEF R(X)
    =INT(X*1ØØ+.5)/1ØØ ::
     DEF S(X)=INT(X*1Ø+.5
    )/1Ø
18Ø FOR Y=1 TO 2 :: FOR X
    =1 TO 6 :: READ P(X,Y
    ):: NEXT X :: NEXT Y
    :: Y=Ø
19Ø FOR X=1 TO 6 :: READ
    P$(X),P(X,3):: NEXT X
2ØØ FOR X=1 TO 7 :: PP(X)
    =X+99 :: NEXT X
21Ø J$="SATSUNMONTUEWEDTH
    UFRI" :: CALL SCREEN(
    12):: FOR X=1 TO 12 :
    : READ F$
22Ø CC$=CC$&RPT$(CHR$(128
    ),5)&F$ :: NEXT X ::
    CC$=CC$&CC$ :: F$=SEG
    $(CC$,LEN(CC$)-8,9)::
     CC$=F$&CC$
23Ø FOR X=1 TO 8 :: READ
    PH$(X):: NEXT X :: FO
    R X=1 TO 22 :: READ H
    C(X):: NEXT X :: GOSU
    B 23ØØ :: GOTO 83Ø
24Ø CC=MT-72Ø :: IF CC<Ø
    THEN CC=CC+K1
25Ø CC=CC/12Ø :: CO=CC-IN
    T(CC):: CC=INT(CC)::
    CO=INT(CO*7+.2):: CC=
    81-(CC*7+CO)
26Ø GOSUB 189Ø :: QQ$=CO$
     :: ROW=16 :: COL=Ø :
    : GOSUB 12Ø
27Ø IF LL>=Ø THEN RETURN
28Ø FOR I=1 TO 16 :: CALL
     GCHAR(16,I,Z):: CALL
     GCHAR(16,33-I,Z1)::
    CALL HCHAR(16,I,Z1)::
     CALL HCHAR(16,33-I,Z
    ):: NEXT I
29Ø FOR I=1 TO 31 :: CALL
     GCHAR(16,I,Z):: IF Z
    =128 THEN 31Ø
3ØØ CALL GCHAR(16,I+1,Z1)
    :: CALL HCHAR(16,1,Z1
    ):: CALL HCHAR(16,I+1
    ,Z):: I=I+1
31Ø NEXT I :: RETURN
32Ø CALL CLEAR :: DISPLAY
     AT(2,9):"** DAYS SKY
     **" :: Q=1 :: GOSUB
    168Ø
33Ø DISPLAY AT(6,1):"INPU
    T THE TIME:" :: DISPL
    AY AT(7,1):"---------
    ------" :: T1,T2=Ø
34Ø DISPLAY AT(9,4):"HOUR
     (Ø-23) ?" :: ACCEPT
    AT(9,18):T1 :: IF T1<
    Ø OR T1>23 THEN Q=1Ø
    :: GOSUB 229Ø :: GOTO
     34Ø
35Ø DISPLAY AT(11,4):"MIN
    UTE (Ø-59) ?" :: ACCE
    PT AT(11,2Ø):T2 :: IF
     T2<Ø OR T2>59 THEN Q
    =12 :: GOSUB 229Ø ::
    GOTO 35Ø
36Ø R$=STR$(T1):: T$=STR$
    (T2):: IF LEN(T$)=1 T
    HEN T$="Ø"&T$
37Ø DISPLAY AT(15,1):"TIM
    E-- ";R$;":";T$
38Ø GOSUB 2Ø5Ø :: IF Z$="
    R" THEN 32Ø
39Ø CALL CLEAR :: T3=T1*6
    Ø+T2+AA-72Ø :: IF T3<
    Ø THEN T3=T3+K1
4ØØ IF T3>K1 THEN T3=T3-K
    1
41Ø MT=T3-36Ø :: IF MT<Ø
    THEN MT=MT+K1
42Ø PT=T3+36Ø :: IF PT>K1
     THEN PT=PT-K1
43Ø DISPLAY AT(1,1):K$;"-
    ";TEM$;STR$(Y);TAB(17
    );STR$(ABS(LL));LL$;"
      ";R$;":";T$;
44Ø CALL COLOR(9,1,5,1Ø,1
    ,5):: TM=VAL(R$&"."&T
    $):: IF TM<6 OR TM>18
     THEN CALL COLOR(9,1,
    2,1Ø,1,2)
45Ø FOR X=2 TO 15 :: CALL
     HCHAR(X,1,1Ø7,32)::
    NEXT X :: XX=7+LC ::
    FOR I=2 TO 32 STEP 2
    :: CALL HCHAR(XX+1,I,
    96):: NEXT I
46Ø GOSUB 24Ø :: ROW=17 :
    : COL=Ø :: IF LL<Ø TH
    EN 49Ø
47Ø IF LL>24 THEN QQ$="E
    {14 SPACES}S
    {15 SPACES}W" :: GOSUB
     12Ø :: GOTO 51Ø
48Ø QQ$="UP-N{6 SPACES}-O
    VERHEAD-{6 SPACES}DOW
    N-S" :: GOSUB 12Ø ::
    GOTO 51Ø
49Ø IF ABS(LL)>24 THEN QQ
    $="W{14 SPACES}N
    {15 SPACES}E" :: GOSUB
     12Ø :: GOTO 51Ø
5ØØ QQ$="UP-S{6 SPACES}-O
    VERHEAD-{6 SPACES}DOW
    N-S" :: GOSUB 12Ø
51Ø T4=AA :: GOSUB 71Ø ::
     Y8=888 :: IF Y9=999
    THEN 55Ø
52Ø Y8=Y9 :: GOSUB 238Ø :
    : IF A1<Ø THEN 55Ø
53Ø IF PK>17Ø3 OR PK<1144
     THEN 55Ø
54Ø GOSUB 11Ø :: IF PKCOL
    >4 AND PKCOL<37 THEN
    CALL HCHAR(PKROW,PKCO
    L-4,97)
55Ø T4=AA+M2*K1 :: IF T4>
    K1 THEN T4=T4-K1
56Ø GOSUB 71Ø :: IF Y9=99
    9 THEN 6ØØ
57Ø MM=INT(M1/9.83333)+1
    :: GOSUB 81Ø :: IF Y9
    =999 THEN 6ØØ
58Ø GOSUB 238Ø :: IF PK>1
    7Ø3 OR PK<1144 THEN 6
    ØØ
59Ø GOSUB 11Ø :: IF PKCOL
    >4 AND PKCOL<37 THEN
    CALL HCHAR(PKROW,PKCO
    L-4,MM):: IF ABS(Y8-Y
    9)<=.5 THEN CALL HCHA
    R(PKROW,PKCOL-4,1Ø8)
6ØØ FOR X=1 TO 7 :: IF X=
    7 THEN 217Ø
61Ø T4=P(X,6):: GOSUB 71Ø
     :: IF Y9=999 THEN 67
    Ø
62Ø U9=SIN(P(X,6)*D9/4)::
     U9=-3*U9+.5 :: U9=IN
    T(U9):: U(X)=U9*4Ø
63Ø PK=1423-Y9+U(X)+L8 ::
     GOSUB 239Ø :: IF PK>
    17Ø3 OR PK<1144 THEN
    67Ø
64Ø GOSUB 11Ø
65Ø IF PKCOL>4 AND PKCOL<
    37 THEN CALL GCHAR(PK
    ROW,PKCOL-4,Z):: IF Z
    <>1Ø7 AND Z<>96 THEN
    PK=PK+1Ø23+SGN(LL)*4Ø
    +(LL=Ø)*4Ø :: GOTO 64
    Ø
66Ø IF PKCOL>4 AND PKCOL<
    37 THEN CALL HCHAR(PK
    ROW,PKCOL-4,PP(X))
67Ø NEXT X :: QQ$="dMERCU
    RY eVENUS fMARS gJUPI
    TER" :: ROW=18 :: COL
    =1 :: GOSUB 12Ø
68Ø QQ$="hSATURN iURANUS
     aSUN  blcMOON" :: RO
    W=19 :: GOSUB 12Ø ::
    QQ$="mNEW MOON + SUN
     " :: ROW=2Ø :: GOSUB
     12Ø
69Ø IF B$<>"" THEN QQ$=B$
     :: ROW=21 :: COL=8 :
    : GOSUB 12Ø
7ØØ QQ$="NEW (T)IME,(P) T
    AB,(D)ATE,(L)AT." ::
    ROW=23 :: COL=Ø :: GO
    SUB 12Ø :: GOTO 181Ø
71Ø Y9=999 :: IF MT<PT TH
    EN 76Ø
72Ø IF T4>=MT OR T4<=PT T
    HEN 74Ø
```

```
730 RETURN
740 IF T4>=MT AND T4<=K1
    THEN 780
750 T4=T4+K1 :: GDTD 780
760 IF T4>=MT AND T4<=PT
    THEN 7B0
770 RETURN
7B0 Y9=INT((T4-MT)/18+.5)
    :: IF Y9=40 THEN Y9=3
    9
790 RETURN
B00 U9=SIN(T4/4*D9):: U9=
    INT(-3*U9+.5)*40 :: R
    ETURN
B10 MM=VAL(SEG$(MM$,3*MM-
    2,3)):: IF LL<0 AND M
    M<>10B THEN MM=197-MM
B20 RETURN
B30 Q=1
B40 CALL CLEAR :: DISPLAY
     AT(2,6):"**** SKYSCA
    PE ****" :: DISPLAY A
    T(4,1):"DATE INPUT" :
    : DISPLAY AT(5,1):"--
    ---------" :: S1=0
B50 IF Y<>0 THEN GDSUB 16
    B0
B60 DISPLAY AT(Q+5,1):"YE
    AR?" :: ACCEPT AT(Q+5
    ,7):Y :: IF Y>=1977 T
    HEN 880
870 OISPLAY AT(Q+5,14):"M
    UST 8E >1977" :: FDR
    I=1 TO 250 :: NEXT I
    :: GOTO 860
8B0 GOSUB 1730 :: DISPLAY
     AT(Q+7,1):"MONTH (1-
    12)?" :: ACCEPT AT(Q+
    7,15):M :: 1F M<1 OR
    M>12 THEN Q=Q+8 :: GO
    SUB 2290 :: Q=Q-8 ::
    GOTD B80
890 DI=VAL(SEG$(MD$,2*M-1
    ,2)):: DI=OI-(M=2)*LY
     :: DI$=STR$(OI)
900 OISPLAY AT(Q+9,1):"DA
    Y (1-";DI$;")?" :: AC
    CEPT AT(Q+9,13):O ::
    IF O<1 OR O>DI THEN Q
    =Q+10 :: GOSU8 2290 :
    : Q=Q-10 :: GOTO 900

910 H$=SEG$(A$,M*3-2,3)
920 OISPLAY AT(Q+11,1):"L
    ATITUOE (-90 TO 90)?"
     :: ACCEPT AT(Q+11,23
    ):LL :: IF A8S(LL)>90
     THEN Q=Q+12 :: GDSU8
     2290 :: Q=Q-12 :: GO
    TO 920
930 GOSUB 2410
940 TEM$=H$&" "&STR$(D)&"
    ," :: OISPLAY AT(Q+14
    ,8):TEM$;Y :: GOSU8 2
    050 :: IF Z$="R" THEN
     Q=4 :: GOTO 840
950 O2=VAL(SEG$(M$,M*3-2,
    3))+O :: GOSU8 1760 :
    : IF M>2 THEN O1=O1+L
    Y :: Y1=Y1+LY
960 O3=D2-1B5 :: IF M=3 A
    NO D<20 THEN O2=O2+LY
     :: D3=O3+LY
970 S5=0 :: IF O3<=0 THEN
     A=180*O2/185 :: GOTO
     990
980 A=1B0*O3/(180+ZY)+180
990 IF A<>1B0 THEN S5=23.
    43333333*SIN(O9*O2*18
    0/185)
1000 IF A>180 THEN S5=-23
     .43333333*SIN(O9*D3)
1010 IF A>=360 THEN A=A-3
     60
1020 A=R(A):: S5=R(S5)::
     A1=(SGN(LL)-(LL=0))*
     S5+90-ABS(LL):: A1=R
     (A1):: GDSUB 13B0 ::
      GDSUB 1330
1030 W=1-(SGN(LL)<0):: IF
      A1>90 THEN A1=1B0-A
     1 :: W=ABS(W-3)
1040 CALL CLEAR :: PRINT
     :: PRINT K$;"-";TEM$
     ;Y;TAB(19);ABS(LL);L
     L$ :: PRINT RPT$("-"
     ,2B)
1050 PRINT :: PRINT "DAY
     DF THE YEAR---  ";ST
     R$(D1):: PRINT :: PR
     INT "SUN'S DATA:": :
1060 PRINT "GEDCENTRIC AN
     GLE--  ";STR$(A);"@"
1070 PRINT "DECLINATION--
     -----  ";STR$(S5);"@
     "
1080 PRINT "ALTITUDE AT N
     DDN--  ";STR$(A1);"@
     ";D1$(W)
1090 PRINT "RIGHT ASCENSI
     DN---  ";A3$
1100 PRINT "R.A. AT 9:00
     PM---  ";A5$ :: PRIN
     T :: PRINT "MODN'S D
     ATA:": :
1110 PRINT "AGE----------
     -----  ";STR$(M1);"
     ";"OY";
1120 PRINT "ELDNGATION---
     -----  ";STR$(M8);"@
     ";L$
1130 PRINT "PHASE -  ";PH
     $(M3): : : :
1140 PRINT "(P)LANET TABL
     E DR NEW (O)ATE" ::
     PRINT :: GOTD 1810
1150 CALL CLEAR :: PRINT
     TAB(6);"** PLANET TA
     8LE **" :: PRINT ::
     PRINT K$;"-- ";TEM$;
     Y;TAB(20);STR$(ABS(L
     L));LL$ :: PRINT ::
     S1=1
1160 PRINT "PLANET  OIST.
      ANG.W/SUN R.A." ::
     PRINT RPT$("-",28):
     :
1170 FDR X=1 TD 6 :: A2=Y
     1/P(X,2)-INT(Y1/P(X,
     2)):: Q3=1
1180 A2=A2*360+P(X,1):: I
     F A2>360 THEN A2=A2-
     360
1190 E=180+A :: IF E>360
     THEN E=E-360
1200 E1=A8S(E-A2):: IF E1
     >180 THEN E1=360-E1
1210 GOSU8 1440 :: E1=E1*
     D9 :: P5=P(X,3):: IF
      X=3 THEN GOSU8 1870
1220 P(X,4)=SQR(1+P5^2-2*
     P5*COS(E1)):: XX=(P5
     ^2-1-P(X,4)^2)/(-2*P
     (X,4))
1230 P(X,5)=-ATN(XX/SQR(-
     XX*XX+1))+PI/2 :: P(
     X,4)=INT(P(X,4)*ES+.
     5):: P(X,5)=P(X,5)/O
     9
1240 P(X,5)=S(P(X,5)):: Q
     1$=STR$(P(X,4)):: Q2
     $=STR$(P(X,5))
1250 Q1=LEN(Q1$):: Q2=LEN
     (Q2$):: GOSU8 1540
1260 PRINT P$(X);TAB(13-Q
     1);Q1$;TA8(20-Q2);Q2
     $;:: IF Q3=-1 THEN P
     RINT "@W";
1270 IF Q3=1 THEN PRINT "
     @E";
1280 GDSUB 1590 :: Q4$=ST
     R$(Q4):: Q5$=STR$(Q5
     ):: IF Q5<10 THEN Q5
     $="0"&Q5$
1290 Q4$=Q4$&":"&Q5$ :: Z
     =LEN(Q4$)
1300 PRINT TAB(22);QQ$;TA
     B(29-Z);Q4$;:: NEXT
     X :: PRINT :: PRINT
     :: PRINT :: PRINT "*
      - VISIBLE AT 9 P.M.
     "
1310 PRINT :: PRINT :: PR
     INT "SUN'S R.A. ----
     ------  ";A3$ :: PRI
     NT "R.A. AT 9:00 P.M
     . ---  ";A5$
1320 PRINT :: PRINT TAB(3
     );"DAYS (S)KY
     {3 SPACES}NEW (D)ATE
     " :: GDTD 1B10
1330 A2=K1*A/360 :: IF A2
     >K1 THEN A2=A2-K1
1340 A3=INT(A2/60):: A4=A
     2-A3*60 :: A5=A3+9 :
     : IF A5>23 THEN A5=A
     5-24
1350 A4=INT(A2-A3*60+.5):
     : IF A4=60 THEN A4=0
      :: A3=A3+1
1360 IF A3=24 THEN A3=0
1370 AA=A3*60+A4 :: GDTO
     1690
1380 M1=(Y1/M9-INT(Y1/M9)
     )*M9+10 :: IF M1>M9
     THEN M1=M1-M9
1390 GDSU8 2080 :: M8=360
     *M2 :: IF M8>180 THE
     N L$="W"
1400 IF M8<=1B0 THEN L$="
     E"
1410 IF M8>180 THEN M8=36
     0-MR
1420 M1=R(M1):: M8=R(M8):
     : YY=INT(7*(Y1/7-INT
     (Y1/7))+.2):: IF YY=
     0 THEN YY=7
1430 K$=SEG$(J$,YY*3-2,3)
     :: RETURN
1440 Q3=0 :: Q1=E+180 ::
     IF Q1>360 THEN 1480
1450 IF A2>E ANO A2<Q1 TH
     EN 1470
1460 Q3=1 :: RETURN
1470 Q3=-1 :: RETURN
1480 Q1=Q1-360 :: IF A2<=
     360 AND A2>E THEN 14
     70
1490 IF Q3<>0 THEN RETURN
1500 IF A2>0 ANO A2<=Q1 T
     HEN 1470
1510 IF Q3<>0 THEN RETURN
1520 IF A2>Q1 THEN 1460
1530 RETURN
1540 Q5=Q3*P(X,5)*4+AA ::
      IF Q5<0 THEN Q5=Q5+
     K1
1550 IF Q5>K1 THEN Q5=Q5-
     K1
1560 P(X,6)=Q5 :: Q4=INT(
     Q5/60):: Q5=INT(Q5-Q
     4*60+.5):: IF Q5=60
     THEN Q5=0 :: Q4=Q4+1
1570 IF Q4=24 THEN Q4=0
15B0 RETURN
1590 SU=A5*60+A4 :: PS=SU
     +360 :: MS=SU-360 ::
      1F PS>K1 THEN PS=PS
     -K1
1600 1F MS<0 THEN MS=MS+K
     1
1610 IF MS>PS THEN 1640
1620 IF P(X,6)<PS AND P(X
     ,6)>MS THEN 1670
1630 QQ$=" " :: RETURN
```

```
1640 IF P(X,6)<K1 AND P(X
     ,6)>MS THEN 1670
1650 IF P(X,6)<PS THEN 16
     70
1660 GOTO 1630
1670 QQ$="*" :: RETURN
1680 OISPLAY AT(Q+3,1):K$
     ;"-- ";TEM$;Y;TAB(20
     );STR$(ABS(LL));LL$;
     :: RETURN
1690 A3$=STR$(A3):: IF A3
     <10 THEN A3$=" "&A3$
1700 A4$=STR$(A4):: IF A4
     <10 THEN A4$="0"&A4$
1710 A3$=A3$&":"&A4$ :: A
     5$=STR$(A5):: IF A5<
     10 THEN A5$=" "&A5$
1720 A5$=A5$&":"&A4$ :: Q
     8=7-LEN(A3$):: Q9=7-
     LEN(A5$):: RETURN
1730 LY=0 :: IF Y/4=INT(Y
     /4)THEN LY=1
1740 IF Y/100=INT(Y/100)A
     NO Y/400=INT(Y/400)A
     NO Y/1000=INT(Y/1000
     )AND Y/4000=INT(Y/40
     00)THEN LY=0
1750 RETURN
1760 Y9=Y+1 :: IF Y9/4=IN
     T(Y9/4)THEN ZY=1
1770 IF Y9/100=INT(Y9/100
     )AND Y9/400<>INT(Y9/
     400)AND Y9/1000=INT(
     Y9/1000)AND Y9/4000=
     INT(Y/4000)THEN ZY=0
1780 Y1=Y-1977 :: Y1=Y1*3
     65+INT(Y1/4)+D1 :: I
     F Y<2000 THEN 1800
1790 Y1=Y1-INT((Y-2001)/1
     00)+INT((Y-2001)/400
     )-INT((Y-1)/4000)
1800 RETURN
1810 CALL KEY(0,KK,SS)::
     IF SS=0 THEN 1810
1820 I$=CHR$(KK):: IF I$=
     "D" THEN Q=4 :: GOTD
      840
1830 IF (I$="S" DR I$="T"
     )AND S1=1 THEN 320
1840 IF I$="P" THEN 1150
1850 IF I$="L" AND S1=1 T
     HEN 2460
1860 GDTD 1810
1870 P5=1.376344086 :: K5
     =A2*4
1880 K5=ABS(K5-1233.73)*9
     0/K1 :: K5=K5*09 ::
     K5=SIN(K5)*.32258122
     4 :: P5=P5+K5 :: RET
     URN
1890 IF CC<=1 THEN CC=CC+
     84
1900 CD$=SEG$(CC$,CC+3,34
     )
1910 IF SEG$(CD$,2,1)<>CH
     R$(128)AND SEG$(CD$,
     3,1)=CHR$(128)THEN C
     D$=SEG$(CD$,1,32)::
     GDTD 1940
1920 IF SEG$(CD$,33,1)<>C
     HR$(128)AND SEG$(CD$
     ,32,1)=CHR$(128)THEN
      CO$=SEG$(CD$,3,32):
     : GDTD 1940
1930 CO$=SEG$(CD$,2,32)
1940 RETURN
1950 OATA 365.26,29.53059
     ,59.818184,42.719626
     ,262.364294,52.91676
     3
1960 OATA 134.69697,218.7
     9464,87.97,224.7,686
     .98
1970 DATA 4332.79813,1075
     9.7195,30686.5884
1980 OATA "MERCURY",.3871
     ,"VENUS",.7233,"MARS
     ",1.5237,"JUPITER",5
     .2028
1990 OATA "SATURN",9.5308
     ,"URANUS",19.182
2000 OATA "SA","SC","LI",
     "VI","LE","CA","GE",
     "TA","AR","PI","AQ",
     "CP"
2010 OATA "NEW","WAXING C
     RESCENT","1ST QUARTE
     R","WAXING GIBBOUS",
     "FULL"
2020 OATA "WANING GIBBOUS
     ","3RO QUARTER","WAN
     ING CRESCENT"
2030 OATA 1770,1719,1620,
     1500,1418,1365,1335,
     1310,1290,1275,1260
2040 OATA 1238,1220,1200,
     1178,1115,915,720,66
     0,640,625,610
2050 OISPLAY AT(20,3):"(R
     )E-INPUT OR (C)DNTIN
     UE"
2060 CALL KEY(0,KK,SS)::
     IF SS=0 THEN 2060
2070 Z$=CHR$(KK):: RETURN
2080 M2=M1/M9 :: IF M1<1
     OR M1>28.5 THEN M3=1
2090 IF M1>=1 AND M1<6.9
     THEN M3=2
2100 IF M1<=8 AND M1>=6.9
      THEN M3=3
2110 IF M1>8 AND M1<14.2
     THEN M3=4
2120 IF M1>=14.2 AND M1<=
     15.2 THEN M3=S
2130 IF M1>15.2 AND M1<21
     .6 THEN M3=6
2140 IF M1>=21.6 AND M1<=
     22.6 THEN M3=7
2150 IF M1>22.6 AND M1<=2
     8.5 THEN M3=8
2160 RETURN
2170 B$="" :: IF Y<>1985
     AND Y<>1986 THEN 670
2180 IF (Y=1985 AND D1<30
     5)DR(Y=1986 AND D1>1
     49)THEN 670
2190 HD=D1+365 :: IF HD>5
     16 THEN HD=HD-365
2200 H1=(HD-295)/10 :: HD
     =INT(H1):: H1=H1-HD
2210 T4=HC(HD)-HC(HD+1)::
      T4=HC(HO)-H1*T4 ::
     IF T4>K1 THEN T4=T4-
     K1
2220 GDSUB 710 :: IF Y9=9
     99 THEN 670
2230 GDSUB 800 :: IF T4>1
     115 AND T4<1200 THEN
      U9=U9+40
2240 IF T4>1290 THEN U9=U
     9-40
2250 IF T4>615 AND T4<=11
     15 THEN U9=U9+80
2260 U(7)=U9 :: B$="jHALL
     EY'S CDMET" :: GDTD
     630
2270 B$=" HALLEY'S CDMET"
2280 GDTO 630
2290 DISPLAY AT(Q,1):DD$
     :: FDR I=1 TD 250 ::
      NEXT I :: CALL HCHA
     R(Q,3,32,14):: RETUR
     N
2300 CALL CHAR(64,"384444
     4438000000",128,RPT$
     ("0",16))
2310 FDR I=0 TD 3 :: READ
      SS :: CALL CHARPAT(
     SS,QQ$):: CALL CHAR(
     I+96,QQ$):: NEXT I
2320 OATA 45,42,41,40
2330 FOR I=0 TO 9 :: REAO
      QQ$ :: CALL CHAR(10
     0+I,QQ$):: NEXT I ::
      CALL COLOR(13,2,9)
2340 OATA 00103B7C7C38100
     0,3C7E66663C187E18,0
     3063C6666663800
2350 OATA 003C427E7E423C0
     0,033E6E76667CC080,0
     0666666624180000,0A15
     2A352A74F8E0
2360 OATA 000000000000000
     0,003C7E7E7E3C00,FFC
     381818181C3FF
2370 RETURN
2380 GOSUB 800 :: PK=1423
     -Y9+U9+LB
2390 IF LL<0 THEN PK=2247
     +80*XX-PK
2400 RETURN
2410 LL$="@N" :: IF LL<0
     THEN LL$="@S"
2420 L1=ABS(LL):: IF ABS(
     LL)<24 THEN L1=40
2430 LC=INT((L1-40)/7+.5)
     :: LB=LC*40 :: O1=VA
     L(SEG$(O$,M*3-2,3))+
     O
2440 IF ABS(LL)<24 THEN L
     B=40*INT(ABS(LL)/7+.
     5)
2450 RETURN
2460 GDSUB 2510 :: DISPLA
     Y AT(6,1):"LATITUDE
     CHANGE" :: DISPLAY A
     T(7,1):RPT$("-",16)
2470 OISPLAY AT(9,1):"INP
     UT NEW LATITUDE:" ::
      ACCEPT AT(9,21):LL
2480 IF ABS(LL)>90 THEN 2
     470
2490 GDSUB 2050 :: IF Z$=
     "R" THEN 2410
2500 GOSUB 2410 :: I$="S"
      :: GDTD 1830
2510 CALL CLEAR :: DISPLA
     Y AT(2,6):"**** SKYS
     CAPE ****" :: Q=1 ::
      GOSUB 1680 :: RETUR
     N
```

©

Crossword Magic

Karen McCullough

Requirements: Apple II-series computer with at least 48K RAM (or Apple III in emulation mode); Apple Macintosh; Commodore 64/128; IBM PC/PCjr with at least 128K RAM; or an Atari 400/800, XL, or XE (memory requirements not available at presstime). All versions require a disk drive, and a printer is recommended. The Apple II version was reviewed; other versions are similar.

Crossword Magic does for the process of creating crossword puzzles what a word processor does for writing. It can't replace the thinking, planning, and research needed to create a satisfying puzzle, but it does simplify the process of organizing and moving the material from brain to paper. *Crossword Magic* lets you create a puzzle on the screen, edit it in various ways, play it, and print it out. The program's authors have provided ways to do everything you can think of with a crossword puzzle.

The program comes on a two-sided disk. One side is called the Maker Disk, and the other the Player Disk. The Maker disk contains the options for creating, editing, printing, deleting, and moving puzzles. The Player disk lets you play a previously created puzzle, or create a new storage disk.

When you start to create a new puzzle, the program first asks if you want automatic sizing. If you answer no, you must enter the size of the grid you desire. However, automatic sizing provides more flexibility, since it allows the grid to grow from its initial size as needed.

Each word you enter is placed in a suitable position on the display grid, highlighted so you always know which word was placed last. Words that don't fit into the grid are added to a list of unused words. If adding a word later allows any unused word to fit into the puzzle, that word is placed on the display and highlighted along with the word just entered. If you don't like where the program placed your word, you can press a key to make the program search for another suitable place, or press another key to remove it.

Menus And Help Screens

A group of special functions also are available at the touch of a key. You can save a partial or complete puzzle; gain access to a help screen that explains your options; return to the main menu (you lose whatever work you've just done on the screen if you don't save it first, however); look at the list of unused words; start entering clues; or go into manual mode. Manual mode lets you add, remove, or change letters in the puzzle.

Crossword Magic comes with a 23-page manual, well-written but not as well organized. Each menu function has its own section in the manual, with clear, comprehensive explanations and directions—until you get to the explanation of the special functions. At that point, each section merely gives you a list of the functions and refers you to a separate section of the manual that explains them in greater detail. The manual would be easier to use if the special functions were explained at the end of each section, even at the expense of some duplication. Also, the special function section begins in the middle of a page, making it difficult to find without referring to the index.

Aside from this, *Crossword Magic* deserves top marks for ease of use, smooth functioning, and good error-handling. It works quickly, finding places for words in seconds, even on large grids. Everything works exactly as described, and the program never failed; it resolutely ignores inappropriate actions. After only a few minutes with the manual, I pulled out a review list of basic Spanish vocabulary words and created a puzzle. However, it's a good idea to read the list of helpful hints in the back of the manual before creating a puzzle; there's a lot of valuable information there.

Crossword Magic is ideal for schools. It's an excellent tool for testing and reinforcing vocabulary in subjects such as English, foreign languages, and science. And anyone who enjoys working with crossword puzzles will find the program a pleasant pastime.

Crossword Magic
Mindscape
3444 Dundee Road
Northbrook, IL 60062
$49.95

Colorasaurus

Steve Hudson

Requirements: Commodore 64 with a disk drive and a joystick; or an Atari 400/800, XL, or XE computer with at least 48K RAM, a disk drive, and a joystick. The Atari version was reviewed.

If you ask a child what makes a good computer game, the answer will probably be that it has to be fun. Ask a parent the same question, and you'll hear words like "enriching" and "educational." But why not get both by creating a game that's captivating enough to hold a child's attention, but stimulating enough to help develop a young mind?

One such game is *Colorasaurus*, an educational program aimed at the three- to six-year-old set. Its goals are straightforward—to help young children develop color discrimination and visual memory skills—and it achieves them with style.

The program actually offers three games in one, and each features lively graphics and ear-catching sound. The first game, "Match," allows the child to match a brightly colored dinosaur (the so-called colorasaurs) with one of three appropriately colored landscapes. Each round presents three new colorasaurs,

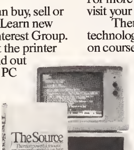
The Source℠ Is Illuminating.

At last, a beacon to help you navigate the vast sea of personal computer technology.

It's The Source. The online information service that can guide you to the best deals in the PC market. Shed light on your software problems. And signal the latest developments in micros.

With The Source, you can buy, sell or swap hardware and software. Learn new applications from a Special Interest Group. See what's been written about the printer you're planning to buy. Or send out an SOS to thousands of other PC users when you need advice.

What's more, The Source can enlighten you with current information on everything from business to travel, investments to world news.



The Source is a service mark of Source Telecomputing Corporation, a subsidiary of The Reader's Digest Association, Inc © 1985 Source Telecomputing Corporation

Whatever kind of help you're after, you can find it faster on The Source. With a system of easy commands and menus. Plus a tutorial—free of online charges—that'll bring you up to speed in a snap.

To sign up today, call **1-800-336-3366.** For more information, mail this coupon, or visit your nearest dealer.

Then, no matter how many waves of new technology rock your boat, you'll stay right on course.

INFORMATION NETWORK
The Source SM
The most powerful resource any personal computer can have.

Show me the light.
Please send more information about The Source.

Name ______________________

7627017

Address ______________________

City, State ______________ Zip ________

Mail to: Source Telecomputing Corporation, P.O. Box 1305, McLean, VA 22102. In Virginia, or outside the U.S., call (703) 821-6666.

COMPUTER AND SOFTWARE SALE

COMMODORE 64 COMPUTER

$139⁹⁵

- 170K Disk Drive $149.95
- Tractor Friction Printer $159.95
- 13" Hi-Res Color Monitor $169.95

CALL BEFORE YOU ORDER

PRICES MAY BE LOWER

C128 COMMODORE COMPUTER

NEW

(Order Now)

$289⁹⁵

- 340K 1571 Disk Drive $259.00
- Voice Synthesizer $39.95
- 12" Green or Amber Monitor $69.95
- Printer/Typewriter Combination $249.95

COMMOOORE 64 COMPUTER $139.95

You pay only $139 95 when you order the powerful 84K COMMODORE 64 COMPUTER! LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your computer that allows you to SAVE OVER $500 off softwore sale prices!! With only $100 of savings opplied your net computer cost is $39 95!!

170K OISK ORIVE $149.95

You poy only $149 95 when you order the 170K Disk Orive! LESS the value of the SPECIAL SOFTWARE OISCOUNT COUPON we pock with your disk drive thot ollows you to SAVE OVER $500 off software sole prices!! With only $100 of savings applied your net disk drive cost is $49 95

13" HI-RES COLOR MONITOR $169.95

You pay only $169 95 when you order this 13 COLOR MONITOR with shorper and clearer resolution than any other color monitors we have tested! LESS value af the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your monitor thot allows you to sove over $500 off softwore sale prices!! With only $100 of savings applied your net color monitar cost is only $69 95 (16 Colors)

80 COLUMN 80 CPS TRACTION/FRICTION PRINTER $159.95

You pay only $159 95 when you order the Comstor T/F deluxe line printer that prints 8½x11 full size single sheet roll or fan fold paper labels etc Impact dot motrix bidirectionol LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your printer thot allows you to SAVE OVER $500 off software sole prices!! With only $100 of savings applied your net printer cost is only $59 95

80 COLUMN BOARO $59.95

Now you program 80 COLUMNS on the screen at one time! Converts your Commodore 64 to 80 COLUMNS when you plug in the 80 COLUMN EXPANSION BOARD!! PLUS 4 slot expander!

80 COLUMNS IN COLOR PAPERBACK WRITER 64 WORD PROCESSOR $39.95

This PAPERBACK WRITER 64 WORD PROCESSOR is the finest available for the COMMODORE 64 computer! The ULTIMATE FOR PROFESSIONAL Word Processing DISPLAYS 40 or 80 COLUMNS IN COLOR or black and white! Simple to operate powerful text editing complete cursor and insert/delete key controls line and paragraph insertion automatic deletion centering margin settings and output to all printers! List $99 00 **SALE $39.95** Coupon $29 95

SPECIAL SOFTWARE COUPON

We pock e SPECIAL SOFTWARE DISCOUNT COUPON with every COMMODORE 64 COMPUTER DISK DRIVE PRINTER or MONITOR we sell! This coupon allows you to SAVE OVER $500 OFF SALE PRICES!!

(Examples)

PROFESSIONAL SOFTWARE COMMODORE 64

Neme	List	Sale	Coupon
Paperback Writer 64	$99 00	$39 95	$29 95
Paperback Data Base	$69 00	$34 95	$24 95
Paperback Dictionary	$24 95	$14 95	$10 00
Practicalc II	$69 95	$49 95	$44 95
Print Shop (Disk)	$44 95	$32 95	$26 95
Practicalc	$59 95	$24 95	$19 95
Programmers Reference Guide	$20 95	$16 95	$12 50
Programmers Helper (Disk)	$59 95	$29 95	$19 95
80 Column Screen (Disk)	$59 95	$29 95	$19 95
Disk Filer (by Flip N File)	$39 95	$14 95	$12 95
Deluxe Tape Cassette	$89 00	$44 95	$34 95
Pro Joy Stick	$19 95	$12 95	$10 00
Light Pen	$39 95	$14 95	$ 9 95
Dust Cover	$ 8 95	$ 6 95	$ 4 60
Simon s Basic	$29 95	$22 95	$19 95
Pitstop II Epyx	$39 95	$22 95	$19 95
Super Graphics Expander	$29 95	$22 95	$19 95
Music Calc 1	$59 95	$29 95	$12 95
Filewriter	$59 95	$29 95	$24 95

(See over 100 coupon items in our cotolog)

Write or call for Sample SPECIAL SOFTWARE COUPON!

ATTENTION Computer Clubs

We Offer Big Volume Discounts

CALL TODAY!

PROTECTO WARRANTY

All Protecto's products carry a minimum 90 day warranty If anything fails within 90 days from the date of purchase simply send your product to us via United Parcel Service prepaid We will IMMEDIATELY send you a replacement at no charge via United Parcel Service prepaid This warranty proves once again that ***We Love Our Customers.***

C128 COMMO OORE COMPUTER $289.95

We expect o limited supply for Christmas We will ship on o first order basis This all-new revolutionary 128K computer uses oll Commodore 64 software and occessories plus all CPM programs formotted for the disk drive
List $349 00 **SALE $289.95.**

340K 1571 COMMODORE OISK ORIVE $259.00

Double Sided Single Disk Drive for C 128 allows you to use C-128 mode plus CPM mode 17 times faster than 1541 plus runs all 1541 formats
List $349 00 **Sele $259.00.**

SUPER AUTO OIAL MOOEM $39.95

Eosy to use Just plug into your Commodore 64 computer ond you re reody to tronsmit ond receive messoges Eosier to use thon dialing your telephone just push one key on your computer! Includes exclusive eosy to use progrom for up ond down loading to printer and disk drives ***Best in U.S.A.***
List $129 00 **SALE $39.95.**

VOICE SYNTHESIZER $39.95

For Commodore 64 computers Just plug it in ond you con program words ond sentences adjust volume ond pitch moke talking odventure games sound oction gomes and customized talkies!! FOR ONLY $19 95 you can add TEXT TO SPEECH just type o word ond hear your computer tolk — ADD SOUND TO ZORK SCOTT ADAMS AND OTHER ADVENTURE GAMES!!
(Disk or tape) List $89 00 **SALE $39.95**

12" GREEN OR AMBER MONITOR $69.95

Your choice of green or amber screen monitor top quolity 80 columns x 24 lines eosy to reod anti-glore! PLUS $9.95 for connecting cable Com-64 or VIC 20 List $159.00 **SALE $69.95.**

PRINTER/TYPEWRITER COMBINATION $249.95

JUKI Superb letter quality doisy wheel printer/typewriter combination Two machines in one — just o flick of the switch 12 extro large corriage typewriter keyboord outomotic margin control ond relocate key drop in cossette ribbon! (90 doy warronty) centronics parallel or RS232 seriol port built in (Specify) List $349 00 **SALE $249.95.** **(Ltd. Qty.)**

13" RGB & COMPOSITE COLOR MONITOR $279.00

Must be used to get 80 columns in color with 80 column computers (C128 & Atori ST)
(Add $14 50 shipping)
List $399 00 **SALE $279.00.**

- **LOWEST PRICES • 15 DAY FREE TRIAL**
- **BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL**

PHONE ORDERS
8 a m - 8 p m Weekdays
9 a m - 12 noon Saturdays

- **90 DAY FREE REPLACEMENT WARRANTY**
- **OVER 500 PROGRAMS • FREE CATALOGS**

* Add $10 00 for shipping handling and insurance Illinois residents please odd 6% tax Add $20.00 for CANADA PUERTO RICO HAWAII, ALASKA APO FPO orders Conodion orders must be in U S dollars WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA
Enclose Cashiers Check Money Order or Personal Check Allow 14 days for delivery 2 to 7 days for phone orders 1 day express mail!
VISA — MASTER CARD — C O D
No C O D to Canada APO FPO

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

240K Apple Compatible Computer System

NEW

SALE

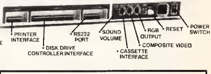
APlus 3000 computer system includes 192K RAM, 48K ROM (32K Microsoft Basic plus 16K ROM Emulator), 144K Laser 5¼" Disk Drive (Runs Apple II Software), Magic Window Wordprocessor, MagiCalc spreadsheet, Magic Memory Database. All for only $499.00

$499.00

Complete System

• 15 Day Free Trial

"Aplus 3000" System



Double Immediate Replacement Warranty

If any of the Aplus 3000 computer system equipment fails due to faulty workmanship or material within 180 days of purchase we will REPLACE it immediately with no service charge!!

• Over 10,000 existing Apple programs • Centronics printer interface included
• 240K (192K RAM, 48K ROM) • ArtSci's Magic Window II, Magic Memory, and MagiCalc included
• 144K Laser 5¼" Disk Drive (Runs Apple II software) • RGB (80 columns in color) and composite included

SPECIFICATIONS

A plus 3000 is a complete, self-contained computer based on the popular 6502A microprocessor and can tap into the tremendous software library of Apple II. Features include 192K Bytes RAM 32KB Enhanced Microsoft BASIC, 80 column text, 560H X 192V color graphic display, 81 key sculptured keyboard and high efficiency switching power supply. Also included as standard are Centronics bus printer interface, Cassette interface 4 channel sound generator, and 5¼" Apple Compatible Disk Drive

• TEXT
- 40 columns X 24 rows or 80 columns X 24 rows software selectable.
- 5 X 7 characters in 7 X 8 matrix
- Upper and lower case characters
- One of Eight colors for characters/graphics and background, Red, Green Blue Cyan Magenta Yellow, Black and White
- Character set with normal, inverse and flashing capabilities

• GRAPHICS
- 280H X 192V 6 colors — Black White, Violet, Green, Blue, Orange.
- 280H X 192V 8 colors bit image — Black White, Red, Green, Blue, Cyan, Magenta Yellow
- 560H X 192V 6 colors — Black, White Violet Green Blue, Orange (High resolution color monitor required)

Super Apple Compatible Disk Drive Sale

Quieter, Cooler, Better Disk Drives for your Apple II plus, IIe, IIc (specify when ordering). List $299.95 Sale $149.95.

15 Day Free Trial — If it doesn't meet your expectations within 15 days of receipt, just send it back to us UPS prepaid and we will refund your purchase price!!

More Features than Apple for less than Commodore

Features	Aplus 3000	Apple IIe	Commodore 128
RAM	**192K**	64K	128K
Runs Apple II Software	**Yes**	Yes	No
Function Keys	**24**	None	16
4 Voice, 6 Octave Sound	**Yes**	No	Yes
Composite Video	**Yes**	Yes	Yes
Disk Drive	**included**	Extra Cost	Extra Cost
Numeric Keypad	**included**	Extra Cost	Included
Video Cable	**included**	Extra Cost	Extra Cost
RGB Color Card	**included**	Extra Cost	Included
80 Column Card	**included**	Extra Cost	Included
Centronics Printer Interface	**included**	Extra Cost	Extra Cost
Drive Controller	**included**	Extra Cost	Included
$150 Wordprocessor (Magic Window)	**included**	Extra Cost	Extra Cost
$150 Spreadsheet (MagiCalc)	**included**	Extra Cost	Extra Cost
$60 Database prg. (Magic Memory)	**included**	Extra Cost	Extra Cost
Your Cost	**$499.00**	$1745.00	$1117.90

ACCESSORIES

	LIST	SALE
2nd Disk Drive	$299.95	$149.95
2 professional analog joysticks	$ 39.95	$ 24.95
Z 80 cart. allows CP/M use	$ 99.95	$ 59.95
RS232 adapter	$ 99.95	$ 59.95
R/F Modulator (TV hookup)	$ 29.95	$ 19.95
RGB cable (RGB Monitor hookup)	$ 24.95	$ 19.95
Centronics cable (for Centronics printer)	$ 34.95	$ 24.95
Technical reference manual	$ 29.95	$ 19.95
Comstar 10x 120-140 CPS dot matrix printer	$399.00	$189.00
80 columns Hi-Res Amber Monitor	$199.00	$ 89.95
80 column Hi Res RGB Monitor	$399.00	$279.00

Add $25.00 for shipping and handling!!

Enclose Cashiers Check Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only. Add $10 more if C.O.D.

COMPUTER DIRECT

We Love Our Customers
22292 N. Pepper Rd., Barrington, Ill. 60010
312/382-5050 to order

and (as the child's responses improve) the three colors become increasingly similar.

The second game, "Find," carries the idea one step further. Like "Match," it asks the child to match colors. However, there are six landscapes instead of three. In addition, it encourages the child to relocate the colorasaurs by recalling which colors were involved. Again, the difficulty increases with the accuracy of the child's responses.

Painting Dinosaurs

The third game, "Colorasaurus," gives the child a chance to personally color a colorasaurus. The child can dip paint from various "paint pots" and then apply it to a large (and by then familiar) colorasaurus that dominates the screen. It's even possible to mix colors or to lighten or darken them (by adding white or black). That gives the child virtually complete control over the resulting colors. The result? Captivated fascination, a great deal of fun, and some worthwhile learning, too.

Each game is controlled with the joystick. Even a young child can move the large, easy-to-see cursor and effectively play any of the games.

The program also uses the keyboard for two special commands. The question mark (?) is a help key that calls up onscreen instructions. Another key returns the player to the main menu. Using either key, it's possible for the child to select various play options—a valuable feature that some educational programs still lack.

Although it's designed for a particular age range, *Colorasaurus* may prove captivating to younger children, too. Although my 17-month-old is too young to manipulate the joystick herself, she loves to sit in my lap and watch the colorasaurs while listening to the dinosaurish music. It's entertaining for older children, too, including us Daddy-types. There's just something about multicolored dinosaurs that appeals to young and old alike.

Colorasaurus
The Learning Company
545 Middlefield Road, Suite 170
Menlo Park, CA 94025
$29.95

Grolier Online Encyclopedia

Dan Gutman

Requirements: Virtually any computer with a modem, telecommunications software, and access to one of 11 major telecommunications services (including CompuServe, The Source, Dow Jones, Dialog, and others).

I just looked up LINCOLN, ABRAHAM in the encyclopedia. There's nothing particularly amazing about that, except that I don't *own* an encyclopedia. With an "electronic encyclopedia" on a mainframe computer that I can tap into anytime I want with my personal computer, I don't need one.

The *Academic American Encyclopedia* from Grolier can be accessed easily on any of 11 different online services by anyone with a modem and a computer. After you log on and hit a few keys, you're dropped into an encyclopedic wonderland of 30,000 articles and 10 million words. Just type SE (for SEarch) and the item you want to look up. The text jumps on the screen in seconds.

An electronic encyclopedia has a few big advantages over a paper one. I can't look up JACKSON, MICHAEL in my parents' old encyclopedia, because he wasn't even born when it was written. Grolier's encyclopedia gets updated every three months. In fact, a week after Leonid Brezhnev died, they had a listing for ANDROPOV, YURI. Also, with the Grolier encyclopedia, I can print out entire articles in seconds on my printer.

On the other hand, while Andropov is covered, there are no listings for LASERDISK, OPTICAL MEMORY, COMPACT DISC, or INTERACTIVE FICTION—terms you'd expect to find in an up-to-date electronic reference source for the 1980s. Michael Jackson gets a paragraph, but you'll find nothing more about recent idols—Prince and Madonna. And the encyclopedia refers to the canceled IBM PCjr as "among the nation's best-selling computers." Of course, any encyclopedia has its limitations.

No Pictures—Yet

There are a few other disadvantages to the Grolier online encyclopedia that are related to its medium. The retrieval commands are picky, so if you misspell a subject you're looking up, the computer may mistakenly tell you there is no listing. For example, if you look up NEWSPAPERS, you'll find nothing. But there *is* a listing for NEWSPAPER. With a printed encyclopedia, you would discover that by flipping through the pages. Also, because of the wide variety of incompatible computers and the limitations of modem communications, the online encyclopedia can't give you the photographs or illustrations you see in a printed encyclopedia.

Someday this may change. Grolier recently announced it is publishing the encyclopedia in the new CD-ROM format (Compact Disc-Read Only Memory). The CD-ROM version, scheduled for release this fall for $199, is quite similar to the online version, except it's stored on a single 4.7-inch compact disc. It requires a special CD-ROM player connected to your computer, such as the one announced last summer by Atari (see "Report from the Summer Consumer Electronics Show" and "Monster Memory," COMPUTE!, August 1985). The CD-ROM encyclopedia has all the search and retrieval features of the online encyclopedia and more—plus it's faster. And although the initial CD-ROM version is text-only, there is plenty of room on the disc to add graphics and digitized illustrations in the future.

Still, even with its current limitations, the Grolier online encyclopedia is worthy of consideration. A conventional encyclopedia might cost $600 or more. On the CompuServe Information Service, Grolier's costs $50 per year plus the regular connect time rates. Depending on how often you access the encyclopedia and how long you stay online, it might take several years before you've spent as much as the conventional encyclopedia would cost. By that time, much of the information in the paper encyclopedia would be out of date and you'd have to buy another one anyway.

If you have school-age children, or if you do a lot of research at home, consider Grolier's online encyclopedia. The convenience of looking things up in seconds is incredible. This is one of the true practical uses for a computer in the home. Besides, think of all the trees you'll save.

Grolier Electronic Publishing
95 Madison Avenue
New York, NY 10016
(Cost varies among information services)

BASIC Lightning & White Lightning For Commodore 64

Roark Dority

Requirements: Commodore 64 and a 1541 disk drive or a tape drive.

You've probably heard of several different software packages which enhance or extend your Commodore 64's BASIC language. *BASIC Lightning*, a graphics development system for the 64, is one of the most exciting such programs I've seen.

BASIC Lightning is much more than a BASIC extension. It's practically a whole new language. Besides all the usual Commodore BASIC commands, *BASIC Lightning* offers more than 150 new commands. They make structured programming possible, let you run up to five parts of a BASIC program simultaneously, and may change your attitude toward using graphics and sound on the 64.

If you've ever programmed in Pascal or a similar language, you'll be happy to know that *BASIC Lightning* includes all the control commands found in Pascal. Control structures include IF–THEN–ELSE, REPEAT –UNTIL, WHILE-WEND, CASE-OF, and procedures and functions with full parameter-passing.

The graphics commands in *BASIC Lightning* are in a class all their own. You can create up to 255 sprites of any size, and these sprites can be scrolled, spun, rotated 90 degrees, enlarged, contracted, and mirrored vertically and horizontally. You can individually design each sprite, place them anywhere on the screen, move part of one sprite into another, copy part of the screen into a sprite, or copy an entire sprite into another.

There are also commands for combining *two* sprites at once in four different ways, and commands to control the sprite colors when two sprites are combined. Another useful feature is the ability to print characters and double-sized characters inside the sprites.

However, I did find it difficult to design sprites with the sprite editor. You can edit only one 8 × 8 grid at a time, and the editor reacts slowly to commands. To design sprites larger than 8 × 8 pixels, the grid must be copied to a larger area on the screen. After several grids have been placed side by side, your sprite begins to take form. Then it's possible to edit more sprites, and even show them in sequence to simulate animation.

Multitasking In BASIC

What *BASIC Lightning* does for graphics, it does for sound as well. For example, music data can be stored in sprites and played in the background with the commands PLAY and RPLAY. This means your music can be playing while the rest of your program is doing other things.

One of the most exciting features of *BASIC Lightning* is its multitasking capability. The TASK command allows up to five things in your program to happen at once. Each task has its own set of variables which are independent of the others. Special commands let you pass values between tasks.

Another product from Oasis Software is *White Lightning*, a Forth-based language. If you have some background in Forth, or are willing to learn a new language, *White Lightning* is certainly a worthwhile package. (Incidentally, *White Lightning* includes *BASIC Lightning*, with all the commands mentioned above.)

BASIC Lightning and *White Lightning* both include a disk and two tapes, so tape users as well as disk users can program with the packages. *BASIC Lightning* is especially ideal for anyone who writes programs in BASIC and is interested in structured programming, sprite graphics, and sound. It's easy to use, too. In minutes it's possible to know enough to handle the screen windows, and everything appears and changes faster than in Commodore BASIC with the POKE commands. *White Lightning* takes longer to learn because it's an entirely different language.

If you're interested in machine language programming, Oasis Software also makes *Machine Lightning*, an advanced machine language system.

Oasis Software
377 Oyster Point Blvd
Unit 15
San Francisco, CA 94080
BASIC Lightning *$39.95*
White Lightning *$49.95*
Machine Lightning *$84.95*

Gato For Apple And IBM

Michael B. Williams

Requirements: Apple IIe or IIc with a disk drive; Apple Macintosh; IBM PC with at least 128K RAM and color/graphics adapter; or an Enhanced Model PCjr. The Apple II version was reviewed; other versions are similar.

Just as flight simulators take the danger out of flying while retaining much of the excitement, *Gato* lets you fight for your country in a submarine from the safety of your desktop. You patrol the South Pacific in your Gato-class submarine, a type of ship actually deployed during World War II.

Your mission begins with a coded message detailing your assignment in enemy territory. At *Gato*'s higher difficulty levels (there are ten), the message is transmitted in Morse code; it's up to you to decipher it. (A Morse code table is included in the manual, but you won't have time to use it without memorizing it first.) You may be ordered to intercept an enemy fleet, cut off enemy supply lines, or rescue allies from enemy territory.

Your patrol area covers 20 allied, enemy, and neutral quadrants of the South Pacific. The patrol chart display offers a view of this entire area, including your sub, the allied subtender, enemy ships, and the area's dozen islands, although not all this information is available on the upper difficulty levels. You can view your position within a quadrant with the quadrant chart, which also shows in greater detail the islands and their surrounding shoals and reefs.

Other displays are the radar screen, the damage report screen, the captain's log (which holds data for eight players), and the main control screen. The damage screen shows a port-side view of your sub, highlighting the damage in any of eight major areas. The main screen demands most of your attention—it contains depth, speed, and heading gauges as well as a full-color view of objects in your area. The Apple version of *Gato* displays these graphics in the extended high-resolution mode; the graphics are adequate, but could be improved.

A nice touch in *Gato* is the fake spreadsheet screen: You can flip to this display to make it look as if you're working whenever the boss strolls by.

Coming Up For Air

Gato promises realism, and it delivers. While the lower difficulty levels are excellent for learning to control the submarine, the upper levels offer extreme challenge and give you no unfair advantages over the enemy as the lower levels do. Attention to detail is very good. You run aground if you get too near an island, and the sub's speed is affected by the ship's depth, the periscope position, and whether the torpedo tube doors are open or closed. Because oxygen is constantly consumed below depths of 20 feet, you must surface occasionally to prevent your crew from suffocating. In addition, depth and speed play a role in how soon you are detected by enemy ships during sneak attacks.

The extensive list of factors the program must calculate and recalculate inevitably slows down the game. The screen updates only about once per second, and takes even longer when ships or islands are nearby.

If one of your torpedos finds its target, you can see the explosions on the display if you're surfaced. The explosion graphics are fair, although the sound effects could use some improvement. Each time you sink a ship, the program updates your captain's log to credit your achievement. The log is reset every time you are sunk—it goes down with the ship.

Gato also includes screens with historic and technical information, plus a demonstration mode (the demo mode explains the submarine but does not show actual game play). The manual moves quickly in an effort to be thorough, including a discussion of strategy and tactics against the five different types of enemy ships. It offers help on attack patterns, defense tactics, avoiding depth charges, and using the radar and periscope.

Gato requires a serious approach if you want to play it well. For those willing to commit themselves to service in the Pacific Fleet, *Gato* lives up to its claims. Just don't expect to sink the entire Japanese fleet on your first (or even fifth) mission.

Gato
Spectrum HoloByte, Inc.
1050 Walnut, Suite 325
Boulder, CO 80302
$39.95 Apple/IBM
$49.95 Macintosh

Atari *PaperClip*

Robert L. Riggs

Requirements: Atari 400/800, XL, or XE with at least 48K RAM, a disk drive, and a printer.

Word processors for Atari computers are reaching an amazing level of sophistication. In many ways, the Atari version of *PaperClip* from Batteries Included is the most sophisticated to date.

Besides all the usual features we've come to expect, *PaperClip* offers a number of capabilities not found in most other Atari word processors. These range from major features such as multiple windows to lesser ones such as character- and word-swap commands. The windows are particularly useful: You can load and edit two different documents simultaneously, and cut and paste text between them.

Typical of the program's flexibility is a configuration menu that lets you customize your own version of *PaperClip*. For instance, you can change the screen background and character colors; choose the screen line length—from 15 to 132 characters—and then determine whether the entire screen window will scroll or just the line being typed; change the left screen margin to correct for TV sets which overscan; elect to use the cursor keys without pressing CTRL; and switch the XL/XE key click and alarm bell on or off. You can even tell *PaperClip* to automatically save the text file you're working on after a predetermined number of keystrokes.

Once you've customized *PaperClip*, you can save it on disk for future use. The program disk isn't copy-protected, so you can make as many backups as you need. You can, for example, create several *PaperClip* disks with different configurations and preferences. To prevent this feature from

being abused by software pirates, *PaperClip* comes with a key that must be plugged into a joystick port to make it work.

The configuration menu offers other choices, too, such as a mini-DOS and options to create, save, and load macro files. A macro is a block of previously defined text—such as a letterhead—that can be placed on the screen with a single keystroke. You can define several macro files, each containing blocks of frequently used text.

PaperClip does not come with a quick reference card for its many commands, but pressing CTRL-SHIFT-? calls up either a disk menu or the online help files (assuming the disk containing these files is inserted in the default drive). The help files contain a list of all *PaperClip* commands necessary for file manipulation, printer control, and screen editing.

Math And Graphics

PaperClip can manipulate numbers and pictures as well as letters. Its built-in calculator can add, subtract, multiply, and divide, printing the answer at the appropriate place in the document. And a screen dump utility on the program disk prints out images created with any of the well-known graphics programs, including the KoalaPad and Atari Touch Tablet or Light Pen. If you want, these pictures can be embedded in your documents, and the program disk contains *B/Graph* and KoalaPad files for practice.

Other useful utilities are included on the disk, too. One program converts *AtariWriter* word processor files to *PaperClip* format. *PaperClip*—like *AtariWriter* and most other Atari word processors—saves text in standard ASCII format, but there are differences between formatting codes and so forth. The conversion utility automatically replaces the *AtariWriter* codes with appropriate *PaperClip* codes.

There's also a mail-merge feature, a typewriter mode which is ideal for addressing envelopes, a word counter, and the ability to search and replace up to six pairs of text strings in a single pass.

One extra feature of *PaperClip* which I especially enjoyed was the rapid cursor movement. The cursor begins repeating sooner when you hold down a cursor key, and it zips across the screen considerably faster than your average Atari cursor.

Versatile Printing

PaperClip is flexible enough to work with virtually any printer. The program disk contains printer drivers for more than 30 of the most popular models. If necessary, you can create your own printer driver by using a program which lets you modify an existing driver or build one from scratch. Therefore, *PaperClip* should be compatible with any future printers.

During my testing, I found that *PaperClip* did not fully support the proportionally spaced font of the Atari 825 printer. *PaperClip* would print the proportional font, but without proportional spacing. However, I was using the early version 1.0 of the program; Batteries Included says the newer version 1.1 does add microspacing for proportional printing, though it still cannot handle true proportional spacing with this printer.

PaperClip has several printing features that will be appreciated for specialized applications—such as a table of contents creator, an option to print any range of pages in a document, the ability to print multiple copies, and a batch-file capability for printing several documents in sequence. It's also the only Atari word processor I've seen that can print in double-column format without forcing you to roll the paper back into the printer—great for newsletters.

Future Features

Because *PaperClip* has such a large number of commands and capabilities, it takes a while to master. The manual is lengthy, and the original edition needs an index and more assistance for first-time users. Batteries Included says a new edition of the manual corrects these deficiencies and adds the much-needed index. It is being shipped with later copies of *Paperclip* 1.1.

Even newer versions of *PaperClip* were scheduled for release this fall. Version 1.2 supports the full 128K RAM in the Atari 130XE, treating the four extra 16K banks as one continuous block of memory. The text area is about 90K long, and the windowing feature lets you load two documents up to 45K long. *PaperClip* 1.2 also will support the extra memory in any future XE models, such as the 256K XE that Atari has hinted about. If this computer ever becomes a reality, *PaperClip* 1.2 would allow more than 200K for text memory.

Batteries Included also planned to make *PaperClip* work with its announced 80-column cartridge, the B.I. 80, but the cartridge was recently canceled due to chip supply problems.

Updates to newer versions of *PaperClip*, incidentally, are available to owners for $10.

Overall, *PaperClip* is without doubt a superb word processor for Atari computers. You won't be sorry you bought it.

PaperClip
Batteries Included
30 Mural Street
Richmond Hill, Ontario
Canada L4B 1B5
$59.95

©

Commodore 64 3-D Animated Graphics

Christian-Marc Panneton

This BASIC enhancement for the Commodore 64 makes it easy to draw and animate impressive three-dimensional graphic figures. While the commands are designed for creating 3-D shapes, they're useful in any high-resolution graphics application.

Nearly everyone has seen three-dimensional computer drawings, but have you ever tried to create one yourself? Since complex math is needed to calculate a 3-D shape and plot it on the high-resolution screen, BASIC takes a long time to draw even relatively simple objects. For this reason, 3-D animation is rarely seen, even in commercial software.

With "3-D Graphics Package," however, you can add several new commands to BASIC for creating sophisticated 3-D graphics—even if you're not a programming wizard.

Type in Program 1 using "MLX," the machine language entry program published elsewhere in this issue. Read the MLX instructions carefully before typing the program, and be sure to save a copy when you're done. Here are the addresses required for MLX:

Starting address: 34000
Ending address: 39381

Because this is a machine language (ML) program, you'll need to load it with LOAD"FILENAME",8,1 for disk or LOAD"FILENAME",1,1 for tape. Activate it by typing SYS 34000 and pressing RETURN. A startup message at the top of the screen reminds you that an enhanced version of BASIC is present. Now type in and save Program 2, a short 3-D graphics demonstration. You *must* activate Program 1 before typing in Program 2. If the enhanced BASIC is not present, the special graphics commands won't work, even if you later reload Program 2 with the enhanced BASIC.

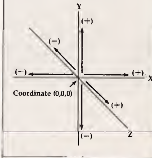
3-D Animation

Program 2 displays two complex, multicolored objects rotating around a common axis. When a rotation is finished, the objects are redisplayed and rotated in a different plane. Notice how short the program is. If you've never tried creating such displays in BASIC, it's difficult to appreciate just how fast and efficient these new commands are. Although objects of this complexity usually take several minutes to draw in BASIC, the ML routines draw and redraw them quickly enough to create a convincing illusion of movement in three-dimensional space.

This program will be easier to use if you understand a few simple concepts. Three-dimensional objects are usually defined in terms of three dimensions or planes relative to you, the observer. The X plane defines horizontal location. The Y plane defines vertical location. The Z plane defines depth. You can locate any point in this system by specifying a *coordinate* for each of the three planes.

As shown in the figure, coordinate (0,0,0) defines the spot where all three planes intersect. In the X plane, negative coordinates lie to the left of the X axis and positive coordinates to the right. In the Y plane, positive coordinates are up and negative ones down. And positive Z coordinates are nearer to you than negative ones.

The 3-D drawing grid is composed of three dimensions or planes. Each point in space has three coordinates on the grid.



Don't worry if that sounds a bit confusing. The best way to learn about these commands is to experiment. Since they all work in direct mode (when you're not running a program), you can type in one command at a time and see the result right away. If it's not what you expect, change one or two values and try again. After a while you'll learn how to draw what you want, even if you're not an expert in geometry.

COMPUTE! Back Issues

Here are some of the applications, tutorials, and games from available back issues of COMPUTE!. Each issue contains much, much more than there's space here to list, but here are some highlights:

May 1981: Named GOSUB/GOTO in Applesoft, Generating Lower Case Text on Apple II, Copy Atari Screens to the Printer, Disk Directory Printer for Atari, Realtime Clock on Atari, PET BASIC Delete Utility, PET Calculated Bar Graphs, Running 40 Column Programs on a CBM 8032, A Fast Visible Memory Dump, Cassette Filing System, Getting to a Machine Language Program, Epidemic Simulation.

June 1981: Computer Using Educators (CUE) on Software Pricing, Apple II Hires Character Generator, Ever Expanding Apple Power, Color Burst for Atari, Mixing Atari Graphics Modes 0 and 8, Relocating PET BASIC Programs, An Assembler in BASIC for PET Quadra PET: Multitasking?, Mapping Unknown Machine Language, RAM/ROM Memory, Keeping Tabs on a Printer.

July 1981: Home Heating and Cooling, Animating Integer BASIC Lores Graphics, The Apple Hires Shape Writer, Adding a Voice Track to Atari Programs, Machine Language Atari Joystick Driver, Four Screen Utilities for the PET, Saving Machine Language Programs on PET Tape Headers, Commodore ROM Systems, Using TAB, SPC, and LEN.

August 1981: Minimize Code and Maximize Speed, Apple Disk Motor Control, A Cassette Tape Monitor for the Apple, Easy Reading of the Atari Joystick, Blockade Game for the Atari, Atari Sound Utility, TI CBM "Fat 40," Keyword for PET, CBM/PET Loading, Chaining, and Overlaying, Adding a Programmable Sound Generator, Converting PET BASIC Programs to ASCII Files.

October 1981: Automatic DATA Statements for CBM and Atari, VIC News, Undeletable Lines on Apple, PET, and VIC; Budgeting on the Apple, Atari Cassette Boot-tapes, Atari Variable Name Utility, Atari Program Library, Train Your PET to Run VIC Programs, Interface a BSR Remote Control System to PET, A General Purpose BCD to Binary Routine, Converting to Fat-40 PET.

December 1981: Saving Fuel $$ (multiple computers), Unscramble Game (multiple computers), Maze Generator (multiple computers), Animating Applesoft Graphics, A Simple Atari Word Processor, Adding High Speed Vertical Positioning to Atari P/M Graphics, OSI Supercursor, A Look at SuperPET, Supermon for PET/CBM, PET Mine Maze Game, Replacing the INPUT # Command, Foreign Language Text on the Commodore Printer, File Recovery.

January 1982: Invest (multiple computers), Developing a Business Algorithm (multiple computers), Apple Addresses, Lowercase with Unmodified Apple, Cryptogram Game for Atari, Superfont: Design Special Character Sets on Atari, PET Repairs for the Amateur, Micromon for PET, Self-modifying Programs in PET BASIC, Tinymon: A VIC Monitor, VIC Color Tips, VIC Memory Map, ZAP: A VIC Game.

May 1982: VIC Meteor Maze Game, Atari Disk Drive Speed Check, Modifying Apple's Floating Point BASIC, Fast Sort for PET/CBM, Extra Atari Colors Through Artifacting, Life Insurance Estimator (multiple computers), PET Screen Input, Getting the Most out of VIC's 500 Bytes.

August 1982: The New Wave of Personal Computers, Household Budget Manager (multiple computers), Word Games (multiple computers), Color Computer Home Energy Monitor, A VIC Light Pen for Under $10, Guess That Animal (multiple computers), PET/CBM Inner BASIC, VIC Communications, Keyprint Compendium, Animation with Atari, VIC Curiosities, Atari Substring Search, PET and VIC Electric Eraser.

September 1982: Apple and Atari and the Sounds of TRON, Commodore Automatic Disk Boot, VIC Joysticks, Three Atari GTIA Articles, Commodore Disk Fixes, The Apple PILOT Language, Sprites and Sound on the Commodore 64, Peripheral Vision Exerciser (multiple computers), Banish INPUT Statements (multiple computers), Charades (multiple computers), PET Pointer Sort, VIC Pause, Mapping Machine Language, Commodore User-defined Functions Defined, A VIC Bug.

January 1983: Sound Synthesis and the Personal Computer, Juggler and Thunderbird Games (multiple computers), Music and Sound Programs (multiple computers), Writing Transportable BASIC, Home Energy Calculator (multiple computers), All About Commodore WAIT, Supermon 64, Perfect Commodore INPUTs, VIC Sound Generator, Copy VIC Disk Files, Commodore 64 Architecture.

May 1983: The New Low-Cost Printer/Plotters, Jumping Jack (multiple computers), Deflector (multiple computers), VIC Kaleidoscope, Graphics on the Sinclair/Timex, Bootmaker for VIC, PET, and 64, VICSTATION: A "Paperless Office," The Atari Musician, Puzzle Generator (multiple computers), Instant 64 Art, 64 Odds and Ends, Versatile VIC Data Acquisition, POP for Commodore.

June 1983: How to Buy the Right Printer, The New, Low-Cost Printers, Astrostorm (multiple computers), The Hawkmen of Dindrin (multiple computers), MusicMaster for the Commodore 64, Commodore Data Searcher, Atari Player/Missile Graphics Simplified, VIC Power Spirals, UnNEW for the VIC and 64, Atari Fast Shuffle, VIC Contractor, Commodore Supermon Q & A.

COMPUTE! Back Issues

July 1983: Constructing the Ideal Computer Game, Techniques for Writing Your Own Adventure Game, SpeedSki and Time Bomb (VIC), Castle Quest and Roadblock (Atari), RATS! and Goblin (64), How to Create a Data Filing System (multiple computers), How to Back Up Disks for VIC and 64, Atari Artifacting, All About the Commodore USR Command, TI Mailing List.

August 1983: Weather Forecaster (multiple computers), First Math and Clues (multiple computers), Converting VIC and 64 Programs to PET, Atari Verify, Apple Bytechanger, VIC and 64 Escape Key, Banish Atari INPUT Statements, Mixing Graphics Modes on the 64, VICplot, VIC/64 Translations: Reading the Keyboard, Musical Atari Keyboard, VIC Display Messages.

September 1983: Games That Teach, Caves of Ice, Diamond Drop, Mystery Spell, and Dots (multiple computers), VIC Pilot, Ultrasort (VIC, 64, PET), Easy Atari Page Flipping, Computer Aided Design on the TI, Relative Files on the VIC/64, Atari Fontbyter, TI Sprite Editor, All About Interrupts (multiple computers), Cracking the 64 Kernal, Making Change on the Timex/Sinclair, Build Your Own Random File Manager (multiple computers).

October 1983: Computer Games by Phone, Coupon File (multiple computers), Dragon Master and Moving Maze (multiple computers), Merging Programs from Commodore Disks, Atari Master Disk Directory, Sprites in TI Extended BASIC, Commodore EXEC, Multicolor Atari Character Editor, High Speed Commodore Mazer, Apple Sounds, Extra Instructions (multiple computers), Commodore DOS Wedges, Invisible Disk Directory for VIC and 64.

February 1984: What Makes a Good Game, Circus (multiple computers), Quatrainment (multiple computers), Commodore 3-D Drawing Master (Apple version also included), Speedy BASIC for VIC and 64, Dr. Video 64.

March 1984: All About Adding Peripherals, Modern Memory: The Future of Storage Devices, Roader (multiple computers), Barrier Battle (multiple computers), Programming the TI: File Processing, Sound Shaper (multiple computers), Commodore Floating Subroutines, Big Buffer for Atari.

April 1984: Apple's Macintosh Unveiled, Securities Analysis (multiple computers), Worm of Bemer (multiple computers), Programming the TI. File Processing, Part 2, 1540/1541 Disk Housekeeping, Hidden Atari DOS Commands, Function Keys for the Apple, TI Tricks and Tips, Super Directory (multiple computers).

May 1984: The Digital Palette: Fundamentals of Computer Graphics, The Inside Story: How Graphics Tablets and Light Pens Work, Picture Perfect for Atari and Commodore 64, 64 Hi-Res Graphics Editor, Snertle (multiple computers), Pentominos: A Puzzle-Solving Program (multiple computers), A BASIC Cross-Reference (PET, 64).

June 1984: Choosing the Right Printer: The Easy Way to Hard Copy, Pests (multiple computers), Olympiad (multiple computers), Programming the TI: TI Graphics, MacroDOS for Atari, Part 1, Apple Variable Save, Programming 64 Sound, Part 1, Apple Input and Menu Screens.

July 1984: Evolutionary to the Core: The Apple IIc Heads for Home, The ABC's of Data Bases, Statistics for Nonstatisticians (multiple computers), Bunny Hop (multiple computers), Blueberries (multiple computers), Atari Artist, Applesoft Lister, Program Conversion with Sinclair BASIC and TI BASIC, Commodore 64 ROM Generations.

September 1984: New Trends in Educational Computing, Choosing the Best Educational Software, Missile Math (multiple computers), Lightsaver (multiple computers), Multiple Choice Quiz Generator (multiple computers), Lightning Sort (multiple computers), Commodore Autoboot, Apple Editing Hints, Atari Paddle Fixer, Musical TI Keyboard.

January 1985: VIC/64 TurboTape: Tape at Disk Speeds, Music in the Computer Age, Inside MSX, Paratrooper (multiple computers), Rescue of Blondell (Commodore/Atari), Guitar Tuner (multiple computers), Which Computer Language Is Best?, Machine Language Multiplication, Part 1, Enhanced Applesoft Input, Atari Terminal Program, IBM Pie Chart Maker.

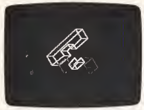
February 1985: Special Games Issue, The New Atari, Fame Games, Birth of a Computer Game, Acrobat (multiple computers), Terminal Program for VIC & 64, Programming the TI Without a Math Background, Adding Sound Effects to Atari, Rebound: Machine Language IBM Game, Apple Bowling Champ, 64 Sound Effects.

Back issues are $6 each. Price includes freight in the US. Outside the US add $1 per magazine ordered for surface postage, $4 per magazine for airmail postage. All back issues are subject to availability.

In the Continental US call
TOLL FREE 800-334-0868
(919-275-9809 in NC)

Or write to:
COMPUTE! Back Issues
P.O. Box 5058
Greensboro, NC 27403 USA

Prepayment required in US funds. MasterCard, Visa, and American Express accepted.
NC residents add 4.5% sales tax.



These figures are redrawn rapidly at different angles to create the illusion of rotation in space.

Following is a description of what each command does. Except for SWAP, every command must be followed by one or more numeric values (numbers or numeric variables).

Large-Scale Commands

These commands are used to prepare the computer for drawing and to perform other general tasks:

SCREEN determines which of three screens is displayed. SCREEN 0 selects the normal text screen. SCREEN 1 switches you to the first graphics screen, and SCREEN 2 displays the second graphics screen. Switching to a graphics screen automatically sets up multicolor high-resolution mode. Animation is simulated by flipping back and forth between the two graphics screens. For instance, you can display a figure on screen 1 while redrawing it on screen 2, then display screen 2 while redrawing the shape on screen 1, and so on. SCREEN 0 restores the text screen when a program is finished.

It's important to remember which screen you're working on. When a graphics screen is displayed, orawing commands appear on that screen. When you're using the text screen, drawing commands take effect on the last graphics screen shown.

Use the function keys f1, f3, and f5 to switch from one screen to another in direct mode. For instance, try pressing f1. The computer prints SCREEN0 followed by a carriage return to execute that command (if you're already in the text screen, nothing changes). Press f3 to perform SCREEN1, f5 to perform SCREEN2, and f1 to return to the text screen. Don't press these keys while a program is running.

DCLEAR clears a graphics screen. Use DCLEAR 1 to erase graphics screen 1 and DCLEAR 2 to clear screen 2. Both screens are cleared when you start up the program.

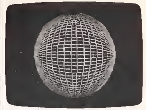
COLOR sets the screen and drawing colors, using color numbers from 0–15 as listed in the 64 manual. This command is followed by five values in the general form COLOR BO,BA,C1,C2,C3. The first two values (BO and BA in this case) set the screen border and background colors. The last three values select drawing colors. In multicolor hi-res mode you can draw in up to three different colors. Thus, COLOR 0,0,1,3,6 sets the border and background colors to black and sets the drawing colors 1, 2, and 3 to white, cyan, and green, respectively. Since drawing commands refer to the drawing colors by number (1, 2, or 3), you should always execute a COLOR command before drawing.

ANGLE is an important command that sets the *observation angles*—your (the observer's) position in space relative to the X-Y-Z grid. Look at the figure again and imagine a cube is drawn there. If you remain stationary and rotate the grid—or if the grid remains stationary and you change your position—the cube's appearance changes. (Since the positioning is relative, you can visualize the change either way.)

ANGLE takes three values, which refer to the Y plane, X plane, and Z plane, respectively. These values represent degrees of rotation around the axis of each plane and must each be in the range −360–360. Program 3 demonstrates a simple use of ANGLE. By redrawing the same shape at different observation angles, you can achieve the illusion of movement in space. Note that ANGLE changes the effect of subsequent drawing commands. It does not change the appearance of existing objects.

PARAM sets four general parameters and should also be used before you begin to draw It takes four values in the general form PARAM X,Y,SC,DI. The first two values (X and Y in this example) locate the *origin* or center of the 3-D grid on the screen. Coordinate (0,0,0) of the grid is located wherever you put the origin. Since the graphics screen contains 160 horizontal pixels (screen dots) and 200 vertical pixels, the X value must be in the range 0–159 and the Y value must be in the range 0–199. Use an X value of 80 and a Y value of 100 to center the origin in the middle of the screen.

The third PARAM value (SC) is *scale*, which controls the overall size of the image. The larger the scale, the bigger the picture, and vice versa. This number must be in the range 0–100; a scale value of 20 works well in many cases. The final PARAM value (DI) is the *distortion* value, a number in the range of 0–250. On most monitors and TVs the pixels are actually wider than they are high, causing a mathematically perfect circle to look elliptical on the screen. This value adds a correction factor to eliminate the distortion. A distortion value of 165 works well in most instances. If your circles still look squashed, experiment with other values.



Only three program lines are needed to draw this spherical surface.

Drawing Commands

These commands draw points, straight lines, and circles or ellipses:

DPLOT draws a point on the current graphics screen and is followed by four values. The first three values set coordinates for the point in Z-X-Y order, and the fourth selects one of the three drawing colors defined in an earlier COLOR command.

DLINE draws a line from one point to another. It requires seven values: three coordinates for the starting point, three coordinates for the the ending point, and the drawing color. Both sets of coordinates are in

Here are 145 reasons to buy at Elek-Tek, not to mention the fastest delivery anywhere.

Save 30% to 43%

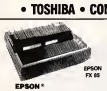
off Manufacturer Suggested Retail prices on

EPSON • Okidata • Star • TOSHIBA • COMREX • OLYMPIA

IOMEGA

BERNOULLI BOX

1. 5 meg Drive for Macintosh ....$ 1375
1a. Cartridges for above 43
2. 10 meg Drive for IBM-PC/XT/AT & compatibles 1790
3. 20 meg Drive for IBM-PC/XT/AT & compatibles 2490
4. 10 meg Drive w/ Auto Boot for IBM PC/XT & compatibles 1890
5. 20 meg Drive w/ Auto Boot for IBM PC/XT & compatibles 2590
6. 10 meg cartridges for above (3 pak special) 125



EPSON®

16.	LX 80	$ 215
17.	RX 100+	350
18.	FX 85	340
19.	FX 185	475
20.	LQ 1500 parallel	CALL
21.	DX10 Daisy Wheel 10CPS	230
22.	DX20 Daisy Wheel 20CPS	CALL
23.	Comrex II Comriter	269

star

45.	SG 10	235
46.	SD 10/15	CALL
47.	SR 10/15	CALL

OKIDATA

48.	Oki 182	CALL
49.	Oki 192/193	CALL
50.	Oki 84	CALL

OTHER PRINTERS

51.	Olympia Compact 2	$ 350
52.	Toshiba P341	CALL
53.	Toshiba P351	CALL
54.	Toshiba P1340	CALL

HEWLETT-PACKARD

73.	HP-71B Computer	$ 399
74.	HP-82400A Series 70 card reader	125
75.	HP-82401A HP IL interface	95
76.	HP-82402A 4K memory module	60
77.	HP-82700A 8K memory module	150
76.	HP-82441A Assembly/Forth	120
79.	HP-82480A HP-71 math pac	75
80.	HP-82482A HP-71 finance pac	80
81.	HP-82483A Surveying Pac	120
82.	HP-82484A Curve Fitting	75
83.	HP-82485A Text Editor	60
84.	HP-82488A Data Com Pac	120

PRINTER BUFFERS

7. **DDP-16** 16K Par/Par (exp. to 64K) ... 70

QUADRAM

8. **QRMPB** 8K Parallel/Parallel 125
9. **QRMPSB** 8K Parallel/Serial 139
10. **QRMSPB** 8K Serial/Parallel 139
11. **QRMSSB** 8K Serial/Serial 139

POWER PROTECTORS

12. **PowerMite PG600-S9** 6 outlet Surge Suppressor 30
13. **ISOBAR 8** 8 outlet Surge Suppressor 50

SAFT

Standby Power Systems

Protects your computer *hardware* and *data* when the power goes out.

14. 400VA (3.3A) 475
    - Transfertime—4-6 Mil Sec
    - Waveform—rectangular
    - Runtime—up to 24 min
15. 1000VA 875
    Switches to battery powered back-up in ½ mil sec.
    - Clean sine wave output
    - AC line synchronized
    - Regulated output voltage
    - Surge Suppression
    - EFI/RFI filter

PROOUCTS FOR IBM-PC

24.	**Amdek 310A** Amber Monitor	$ 150
25.	**Generic Multi** Multifunction Board, 64K	150
26.	**AST Six Pak +** Multifunction Board, 64K	229
27.	**AST Six Pak + (loaded)** Multifunction Board, 384K	289
28.	**AST Megaplus II** Multifunction Board, 64K	270
29.	**Quadram Quadboard** Multifunction Board, 0K	195
30.	**Quadram Quadboard** Multi Board 64K/384K	210/267
31.	**Orchid Tech.** PC Turbo 186	570
32.	**Paradise** Modular Graphics Card	275
33.	**Hercules** Monochrome Card	329
34.	**Hercules Color** Color Graphic Card	170
35.	**Novation 490605-1** 2400BPS inc Mite Software	620
38.	**Novation 490603** ½ Card Modem 2400BPS No software	425
37.	**Novation 490603-1** As above inc MS-DOS Software	490
38.	**Hayes 1200B** Internal modem w/software	379
39.	**AT&T 4000** 300/1200 Ext Modem	335
40.	**Hayes 1200** External modem	399
41.	**Hayes 2400** External modem	599
42.	**US Robotics Courier 2400** Ext 2400B Smart Modem	460
43.	**US Robotics Telpac** Telecomm Software	75
44.	**TEAC FD55B** ½ ht DSDD Disk Drive	90

PROOUCTS FOR APPLE

55.	**U—Print IIc-16K** Par Interface external	$ 85
58.	**Grappler PLUS** Graphics Interface	80
57.	**Buffered Grappler** 16K Buffer (Expand to 64K)	150
58.	**Kensington System Saver** Fan and Surge Suppr	69
59.	**Super Cooling Fan** Fan and Surge Suppr	29
60.	**Gamma** Disk Controller Card	48
61.	**Gamma II** SS Disk Drive for II, IIe, IIplus	125
62.	**Gamma IIc** SS Disk Drive for IIc	125
63.	**Rana ELITE 2** DSSD Disk Drive	335
64.	**Rana ELITE 3** DSDD Disk Drive	420
65.	**ERAM 80** 80 col, 64K RAM for IIe	115
66.	**Novation Apple Cat II** 300 Baud Modem	200
67.	**Novation 212** 300/1200 Baud Modem	390
68.	**Nov 49059112** Mod. & Soft. for Macintosh	CALL
69.	**Hayes Micromodem IIe** 300 Baud Modem	145
70.	**Hayes Smartmodem IIc** 300 Baud Modem for IIc	240
71.	**Wico 501030** Analog Joystick	36
72.	**Amdek Monitors**	CALL
73.	**Zenith Monitors**	CALL

SPECIAL PURCHASE

HP-41C $105

While Limited Quantities Last

86.	HP-41CV	$166
87.	HP-41CX	245
88.	Optical Wand	95
89.	Card Reader	145
90.	Printer	283
91.	Quad R A M (for HP41C)	60
92.	Ext. Memory Module	60
93.	Ext. Function Module	60
94.	Time Module	80
95.	HP-IL Loop Module	95
96.	Digital Cassette Drive	400
97.	Printer/Plotter (HP IL)	335
98.	HP-2225B ThinkJet Printer	375
99.	HP-9114A Disk Drive	600

SLIMLINE Shirtpocket Styled Power Packed Programmable LCD PROBLEM SOLVERS

100.	HP-11C Scientific	58
101.	HP-15C Scientific	90
102.	HP-12C Financial	90
103.	HP-16C Programmer	90

PERSONAL COMPUTATION DEALER

HUGE SAVINGS ON DISKETTES

	Dysan	maxell	3M	SONY	wabash	MEMOREX
3½" SSDD	28.00	25.00	25.00	25.00	—	—
DSDD	37.00	37.00	37.00	37.00	—	—
5¼" SSDD	17.00	14.00	14.50	13.00	11.50	11.50
DSDD	21.00	18.00	18.00	16.00	12.50	14.00
SSDD96TPI	28.00	24.00	24.00	—	—	—
DSDD96TPI	37.00	29.00	29.00	—	—	—
5¼" DSDDHD (For IBM AT)	45.00	39.00	40.00	—	24.00	—
8" SSDD**	25.00	29.00	25.00	—	19.00	—
8" DSDD**	29.00	32.00	29.00	—	20.00	—

**Unformatted

Call for Quantity pricing for 10 boxes or more.

3M DATA CARTRIDGES

140. OC100 ....$ 14.00
141. OC300A 18.00
142. OC300XL 21.00
143. OC300XL/P 22.00
144. OC600A 23.50
145. OC1000 ....15.00

Call for Quantity pricing for 10 cartridges or more

CALL TOLL FREE 800-621-1269 EXCEPT Illinois, Alaska

ELEK-TEK, inc. 6557 N. Lincoln Ave. Chicago, IL 60645 (312) 631 7800 (312) 677 7660

Corp. Accts Invited. Min. Ord. $15.00 Visa or MasterCard by Mail or Phone. Mail Cashier's Check, Mon. Ord., Personal Check (2 wks. to clear). Add $4.00 1st item. (AK, HI, PR, Canada add $10.00 first item) $1.00 ea. add'l shpg. & handl. Shipments to IL address add 7% tax. Prices subj. to change. WRITE for free catalog. RETURN POLICY: Defectives Only. Most products replaced within 30 days of purchase with identical merchandise only. Computer and large peripherals replaced only when defective on arrival (within 3 work days of delivery). Other problems covered by mfr. warranty. ALL ELEKTEK MERCHANDISE IS BRAND NEW, FIRST QUALITY AND COMPLETE. Delivery subject to availability. DUNS #09-718-0517

Z-X-Y order. The following program demonstrates a simple use of DLINE:

```
10 COLORØ,Ø,1,1Ø,3:PARAM8Ø,1ØØ
   ,2Ø,165:DCLEAR1:SCREEN1:FOR
   J=ØTO36ØSTEP5
2Ø ANGLEJ,9Ø,Ø:DLINE35ØØ,Ø,Ø,4
   5ØØ,Ø,Ø,3:NEXT
```

Press f1 to return to screen 0 when the program is finished. Change the 90 in line 20 to 12 and run it again to see how a different observation angle affects the object's appearance.

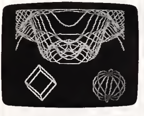
DDRAW works like DLINE but starts drawing at the point where a previous DPLOT, DLINE or DDRAW command left off. Since the beginning point is already defined, this command needs only four values: three coordinates for the ending point and a drawing color. For example, DDRAW −500,0,0,1 draws a line from the previous point to (−500,0,0) in color 1.

DCIRCLE draws a circle or ellipse and requires eight values. The first three values are Z-X-Y coordinates that define the center of the circle. The fourth value sets the *radius*, or distance from the center to the circle's edge. The next three values define orientation angles for the circle, and the last value sets the drawing color.

This command takes the general form DCIRCLE Z,X,Y,R,AY,AZ,AX,C. As with ANGLE, the DCIRCLE orientation angles control which way the circle faces. When all three angles are zero, the circle is drawn in the Z-Y plane. Increasing the value of AY causes a counterclockwise rotation around the Y axis. If AY is 90, AZ is 0, and AX is 0, the circle is drawn in the X-Y plane. Increasing the value of AZ rotates the circle counterclockwise around the Z axis. When AY is 0, AZ is 90, and AX is 0, the circle is drawn in the X-Z plane. Increasing the value of AX rotates the circle counterclockwise around the X axis.

DCIRCLE uses integer computations to speed up the drawing process. Though the command accepts noninteger (fractional) values, it only uses the integer part of the number. This program shows how a spherical surface can be formed out of many circles.

```
1Ø COLORØ,Ø,1,1Ø,3:PARAM8Ø,1ØØ
   ,2Ø,165:DCLEAR1:SCREEN1
2Ø FORJ=9ØTO ØSTEP-5:ANGLEØ,J,
   Ø:DCIRCLEØ,Ø,Ø,45ØØ,Ø,9Ø,Ø,
   2:NEXT
3Ø FORJ=5TO75STEP1Ø:ANGLEØ,9Ø,
   Ø:DCIRCLEØ,Ø,Ø,45ØØ,Ø,J,Ø,3
   :NEXT
```



"3-D Graphics Package" helps you draw complex shapes like these.

Animation Commands

This group of commands simplifies the process of drawing and redrawing complex objects:

ANIM stands for *animate* and takes one value corresponding to the screen you want to affect. ANIM 1 displays graphics screen 2, clears screen 1, and lets you draw on screen 1. ANIM 2 does the reverse: Screen 1 is displayed, screen 2 is cleared, and you're ready to draw on screen 2. Program 2 demonstrates a typical use of ANIM.

SWAP exchanges the contents of screen 1 and screen 2, providing another means of animation. For instance, you might display screen 1 at all times, redrawing the figure on screen 2 (which is not seen), then quickly move the new figure into screen 1 with SWAP. This command requires no parameters.

FSET is a very powerful command that lets you define up to three figures. Once a figure is defined, it can be drawn quickly at any time with a FIGURE command (see below). A figure consists of a series of drawing instructions, and each use of FSET lets you add one drawing instruction to the figure.

The general form of the command is FSET FN,Z,X,Y,C,I. In this example, FN sets the figure number that determines which of the three possible figures you are working on. Z, X, Y, and C represent three coordinates and a drawing color, and I represents the drawing instruction. The instruction can be either a DPLOT or a DLINE command. If I is 0, then FSET performs DPLOT, drawing a point at (Z,X,Y) in the color C. If I is 1, FSET performs DLINE, drawing a line from the last coordinate defined to the point (Z,X,Y) in the color C. The first of the three figures defined by FSET may contain up to 120 separate drawing instructions. Figures 2 and 3 are limited to 80 instructions each.

FIGURE is used to draw an entire figure previously defined with an FSET command. It takes a single value corresponding to the figure number. For instance, FIGURE 1 draws the first figure defined with FSET. FIGURE 2 draws the second, and so on.

FCLEAR clears any of the three figure definitions, permitting you to create new figures with FSET. FCLEAR 1 clears the figure 1, FCLEAR 2 clears the figure 2, and so on.

Memory Allocation

Here are the various memory areas used by this program:

32768–33791	Screen 2 color memory
33792–40959	Program code
40960–49151	Screen 2 bitmap
49152–52223	Figure definitions
52224–53247	Screen 1 color memory
57344–65535	Screen 1 bitmap

Quick Reference Table

ANGLE Y,X,Z
ANIM N
COLOR BO,BA,C1,C2,C3
DCIRCLE Z,X,Y,R,AY,AZ,AX,C
DCLEAR N
DDRAW Z,X,Y,C
DLINE Z,X,Y,Z1,X1,Y1,C
DPLOT Z,X,Y,C
FCLEAR FN
FIGURE FN
FSET FN,Z,X,Y,C,I
PARAM X,Y,SC,DI
SCREEN N
SWAP

Program 1: 3-D Graphics Package

Please refer to the "MLX" article in this issue before entering the following listing

```
34000 :032,189,137,032,193,137,160
34006 :032,015,153,032,036,153,123
34012 :032,055,153,169,000,141,002
34018 :084,003,141,085,003,141,171
34024 :086,003,141,087,003,141,181
34030 :088,003,141,089,003,032,082
34036 :101,141,169,080,141,099,207
34042 :003,169,100,141,100,003,254
34048 :169,010,141,101,003,169,081
```

```
34054 :165,141,102,003,169,160,234
34060 :141,078,138,032,068,166,123
34066 :169,012,141,064,003,169,064
34072 :011,141,060,003,169,010,162
34078 :141,061,003,169,013,141,046
34084 :062,003,169,014,141,063,232
34090 :003,032,117,137,032,091,198
34096 :137,024,162,255,160,127,145
34102 :032,153,255,032,191,227,176
34108 :120,169,019,141,004,003,004
34114 :169,136,141,005,003,169,177
34120 :201,141,006,003,169,135,215
34126 :141,007,003,169,163,141,190
34132 :008,003,169,135,141,009,037
34138 :003,169,228,141,024,003,146
34144 :169,133,141,025,003,169,224
34150 :001,141,022,003,169,134,060
34156 :141,023,003,169,074,141,147
34162 :143,002,169,153,141,144,098
34168 :002,088,160,000,185,138,181
34174 :133,032,210,255,200,192,124
34180 :090,208,245,076,116,164,007
34186 :147,153,018,032,032,032,040
34192 :051,045,068,073,077,069,015
34198 :078,083,073,079,078,032,061
34204 :080,073,067,084,085,082,115
34210 :069,032,065,078,065,076,035
34216 :089,083,073,083,032,032,048
34222 :032,032,032,032,032,032,110
34228 :032,146,154,153,018,032,203
34234 :032,032,032,032,032,032,122
34240 :032,032,032,066,089,032,219
34246 :067,072,082,073,083,084,147
34252 :073,065,078,032,077,046,063
34258 :032,080,065,078,078,069,100
34264 :084,079,078,032,032,032,041
34270 :032,032,032,146,154,013,119
34276 :072,138,072,152,072,169,135
34282 :127,141,013,221,172,013,153
34288 :221,016,003,076,114,254,156
34294 :032,188,246,032,225,255,200
34300 :240,003,076,114,254,165,080
34306 :055,133,001,032,163,253,127
34312 :032,024,229,032,018,133,220
34318 :108,002,160,230,122,208,076
34324 :002,230,123,096,032,017,008
34330 :134,032,138,173,032,191,214
34336 :177,032,121,000,164,100,114
34342 :165,101,096,162,000,165,215
34348 :100,016,015,162,001,056,138
34354 :169,000,229,101,133,101,015
34360 :169,000,229,100,133,100,019
34366 :096,032,027,134,032,041,168
34372 :134,142,092,003,165,100,192
34378 :141,091,003,165,101,141,204
34384 :090,003,032,024,134,032,139
34390 :041,134,142,095,003,165,154
34396 :100,141,094,003,165,101,184
34402 :141,093,003,032,024,134,013
34408 :032,041,134,142,098,003,042
34414 :165,100,141,097,003,165,013
34420 :101,141,096,003,076,000,021
34426 :144,032,063,134,165,038,186
34432 :133,251,133,025,165,040,107
34438 :133,252,133,026,032,024,222
34444 :134,165,101,141,065,003,237
34450 :076,108,138,032,063,134,185
34456 :165,038,133,251,165,040,176
34462 :133,252,032,017,134,032,246
34468 :063,134,165,038,133,253,182
34474 :165,040,133,254,032,024,050
34480 :134,141,065,003,076,242,069
34486 :138,032,027,134,141,099,241
34492 :003,032,024,134,141,100,110
34498 :003,032,024,134,141,101,117
34504 :003,032,024,134,141,102,124
34510 :003,096,032,027,134,208,194
34516 :003,076,091,137,201,001,209
34522 :208,011,169,224,141,078,025
34528 :138,032,117,137,076,039,251
34534 :137,169,160,141,078,138,029
34540 :032,117,137,076,065,137,032
34546 :032,027,134,201,001,208,077
34552 :011,032,039,137,169,160,028
34558 :141,078,138,076,193,137,249
34564 :032,065,137,169,224,141,004
34570 :078,138,076,189,137,032,148
34576 :027,134,141,064,003,032,161
34582 :024,134,141,060,003,032,160
34588 :024,134,141,061,003,032,167
34594 :024,134,141,062,003,032,174
34600 :024,134,141,063,003,076,225
34606 :117,137,032,027,134,141,122
34612 :084,003,140,085,003,032,143
34618 :024,134,141,086,003,140,074
34624 :087,003,032,024,134,141,229
34630 :088,003,140,089,003,076,213
34636 :101,141,032,027,134,140,139
34642 :156,003,141,155,003,032,060
34648 :024,134,140,158,003,141,176
34654 :157,003,032,024,134,140,072
34660 :160,003,141,159,003,032,086
34666 :024,134,140,162,003,141,198
34672 :161,003,032,024,134,140,094
34678 :085,003,141,084,003,032,210
34684 :024,134,140,087,003,141,141
34690 :086,003,032,024,134,140,037
34696 :089,003,141,088,003,032,236
34702 :024,134,141,065,003,076,073
34708 :251,145,032,027,134,201,170
34714 :001,208,003,076,189,137,000
34720 :076,193,137,032,115,000,201
34726 :201,204,144,025,201,217,134
34732 :176,021,032,180,135,076,024
34738 :174,167,233,203,010,168,109
34744 :185,014,137,072,185,013,022
34750 :137,072,076,115,000,032,110
34756 :121,000,076,231,167,016,039
34762 :066,201,255,240,062,036,038
34768 :015,048,058,170,132,073,192
34774 :201,204,176,010,160,160,101
34780 :132,035,160,158,132,034,103
34786 :208,011,233,076,170,160,060
34792 :136,132,035,160,199,132,002
34798 :034,160,000,010,240,016,186
34804 :202,016,012,230,034,208,178
34810 :002,230,035,177,034,016,232
34816 :246,048,241,200,177,034,178
34822 :048,008,032,071,171,208,032
34828 :246,076,243,166,076,239,034
34834 :166,166,122,160,004,132,000
34840 :015,189,000,002,016,007,253
34846 :201,255,240,062,232,208,204
34852 :244,201,032,240,055,133,173
34858 :008,201,034,240,086,036,135
34864 :015,112,045,201,063,208,180
34870 :004,169,153,208,037,201,058
34876 :048,144,004,201,060,144,149
34882 :029,132,113,160,000,132,120
34888 :011,136,134,122,202,200,109
34894 :232,189,000,002,056,249,038
34900 :158,160,240,245,201,128,192
34906 :208,048,005,011,164,113,127
34912 :232,200,153,251,001,185,094
34918 :251,001,240,089,056,233,204
34924 :058,240,004,201,073,208,124
34930 :002,133,015,056,233,085,126
34936 :208,159,133,008,189,000,049
34942 :002,240,223,197,008,240,012
34948 :219,200,153,251,001,232,164
34954 :208,240,166,122,230,011,091
34960 :200,185,157,160,016,250,088
34966 :185,158,160,208,180,160,177
34972 :255,202,200,232,189,000,210
34978 :002,056,249,199,136,240,020
34984 :245,201,128,208,002,240,168
34990 :173,166,122,230,011,200,052
34996 :185,198,136,016,250,185,126
35002 :199,136,208,226,189,000,120
35008 :002,016,155,076,009,166,104
35014 :032,068,080,076,079,212,233
35020 :068,076,073,078,197,068,252
35026 :068,082,065,215,080,065,017
35032 :082,065,205,083,067,082,032
35038 :069,069,206,065,078,073,014
35044 :205,067,079,076,079,210,176
35050 :065,078,071,076,197,068,021
35056 :067,073,082,067,076,197,034
35062 :068,067,076,069,065,210,033
35068 :070,083,069,212,070,073,061
35074 :071,085,082,197,070,067,062
35080 :076,069,065,210,000,122,038
35086 :134,148,134,162,134,182,140
35092 :134,207,134,241,134,014,116
35098 :135,047,135,077,135,149,192
35104 :135,078,151,057,152,007,100
35110 :153,173,000,221,041,248,106
35116 :009,004,141,000,221,169,076
35122 :057,141,024,208,169,059,196
35128 :141,017,208,169,024,141,244
35134 :022,208,096,173,000,221,014
35140 :041,248,009,005,141,000,000
35146 :221,169,009,141,024,208,078
35152 :169,059,141,017,208,169,075
35158 :024,141,022,208,096,173,238
35164 :000,221,041,248,009,007,106
35170 :141,000,221,169,021,141,023
35176 :024,208,169,027,141,017,178
35182 :208,169,008,141,022,208,098
35188 :096,173,064,003,141,032,113
35194 :208,173,060,003,141,033,228
35200 :208,173,061,003,010,010,081
35206 :010,010,013,062,003,160,136
35212 :250,153,255,203,153,249,123
35218 :204,153,243,205,153,237,061
35224 :206,153,255,127,153,249,015
35230 :128,153,243,129,153,237,177
35236 :130,136,208,229,173,063,079
35242 :003,160,250,153,255,215,182
35248 :153,249,216,153,243,217,127
35254 :153,237,218,136,208,241,095
35260 :096,169,224,208,002,169,032
35266 :160,133,088,169,224,133,077
35272 :089,169,000,133,087,168,078
35278 :145,087,200,208,251,230,047
35284 :088,230,089,208,245,096,144
35290 :165,026,074,074,074,133,252
35296 :087,010,010,024,101,087,031
35302 :133,087,074,074,133,088,051
35308 :165,087,010,010,010,010,016
35314 :010,010,133,087,169,007,146
35320 :037,026,024,101,087,133,144
35326 :087,096,165,025,074,074,007
35332 :133,089,074,074,074,074,010
35338 :074,133,090,165,089,010,059
35344 :010,010,133,089,169,003,174
35350 :037,025,170,169,000,133,044
35356 :091,173,065,003,240,024,112
35362 :106,144,005,189,104,138,208
35368 :133,091,173,065,003,106,099
35374 :106,144,009,024,189,104,110
35380 :138,010,101,091,133,091,104
35386 :096,152,072,120,169,048,203
35392 :133,001,024,165,089,101,065
35398 :087,133,093,165,090,101,227
35404 :088,105,224,133,094,160,112
35410 :000,177,093,061,100,138,139
35416 :005,091,145,093,169,055,134
35422 :133,001,088,104,168,096,172
35428 :063,207,243,252,064,016,177
35434 :004,001,032,218,137,032,018
35440 :000,138,076,059,138,198,209
35446 :028,056,165,101,229,099,028
35452 :133,101,165,028,229,100,112
35458 :133,028,032,000,138,164,113
35464 :026,200,136,132,026,032,176
35470 :218,137,032,059,138,196,154
35476 :028,208,243,096,032,000,243
35482 :138,164,026,136,200,132,182
35488 :026,032,218,137,032,059,152
35494 :138,196,028,208,243,096,051
35500 :230,027,024,165,101,101,052
35506 :099,133,101,165,027,101,036
35512 :100,133,027,032,218,137,063
35518 :164,025,136,200,132,025,104
35524 :032,000,138,032,059,138,083
35530 :196,027,208,243,096,198,146
35536 :027,056,165,101,229,099,117
35542 :133,101,165,027,229,100,201
35548 :133,027,032,218,137,164,163
35554 :025,200,136,132,025,032,008
35560 :000,138,032,059,138,196,027
35566 :027,208,243,096,165,251,204
35572 :133,025,165,252,133,026,210
35578 :197,254,144,012,032,039,160
35584 :139,165,253,133,251,165,082
35590 :254,133,252,096,165,253,135
35596 :133,251,165,025,133,253,204
35602 :165,251,133,025,165,254,243
35608 :133,252,165,026,133,254,219
35614 :165,252,133,026,056,032,182
35620 :039,139,096,229,254,133,158
35626 :102,165,251,197,253,144,130
35632 :060,229,253,197,102,144,009
35638 :027,133,098,165,102,133,200
35644 :097,032,174,139,165,251,150
35650 :133,027,230,027,032,207,210
35656 :138,198,026,165,027,197,055
35662 :253,208,245,096,133,097,086
35668 :165,102,133,098,032,174,020
35674 :139,165,252,133,028,230,013
35680 :028,032,117,138,198,025,122
35686 :165,028,197,254,208,245,175
```

```
35692 :096,056,165,253,229,251,134
35698 :197,102,144,029,133,098,049
35704 :165,102,133,097,032,174,055
35710 :139,230,099,165,251,133,119
35716 :027,198,027,032,172,138,214
35722 :198,026,165,027,197,253,236
35728 :208,245,096,133,097,165,064
35734 :102,133,098,032,174,139,060
35740 :165,252,133,028,230,028,224
35746 :032,117,138,230,025,165,101
35752 :028,197,254,208,245,096,172
35758 :230,097,230,098,162,008,231
35764 :165,098,133,100,169,000,077
35770 :006,100,042,197,097,144,004
35776 :004,229,097,230,100,202,030
35782 :208,242,133,098,162,008,025
35788 :169,000,133,099,165,098,100
35794 :006,099,042,197,097,144,027
35800 :004,229,097,230,099,202,053
35806 :208,242,198,100,169,000,115
35812 :133,101,096,000,000,001,047
35818 :030,002,060,003,089,004,166
35824 :119,005,148,006,177,007,190
35830 :205,008,232,010,003,011,203
35836 :029,012,054,013,078,014,196
35842 :102,015,124,016,144,017,164
35848 :164,018,182,019,199,020,098
35854 :214,021,228,022,240,023,250
35860 :250,025,002,026,008,027,102
35866 :012,028,014,029,014,030,153
35872 :012,031,007,032,002,033,146
35878 :246,033,234,034,219,035,071
35884 :202,036,181,037,158,038,184
35890 :132,039,103,040,071,041,220
35896 :035,041,253,042,211,043,169
35902 :166,044,117,045,065,046,033
35908 :010,046,206,047,144,048,057
35914 :212,049,007,049,189,050,118
35920 :111,051,029,051,199,052,061
35926 :109,053,015,053,173,054,031
35932 :070,054,220,055,109,055,143
35938 :250,056,130,057,006,057,142
35944 :134,058,001,058,120,058,021
35950 :234,059,087,059,192,060,033
35956 :036,060,131,060,222,061,174
35962 :052,061,133,061,210,062,189
35968 :025,062,092,062,154,062,073
35974 :211,063,007,063,054,063,083
35980 :097,063,134,063,166,063,214
35986 :194,063,216,063,234,063,211
35992 :246,063,254,064,000,239,250
35998 :132,238,231,246,254,255,234
36004 :246,231,132,143,149,142,183
36010 :148,150,142,134,145,242,107
36016 :146,134,149,132,231,151,095
36022 :134,145,145,138,150,144,014
36028 :145,133,106,132,107,169,212
36034 :000,157,068,003,157,071,138
36040 :003,152,016,013,024,169,065
36046 :104,101,106,133,106,144,132
36052 :002,230,107,230,107,056,176
36058 :165,106,233,180,168,165,211
36064 :107,233,000,144,005,132,077
36070 :106,254,068,003,165,106,164
36076 :201,090,144,014,189,068,174
36082 :003,208,003,254,071,003,016
36088 :169,180,229,106,176,008,092
36094 :189,068,003,157,071,003,233
36100 :165,106,010,133,106,168,180
36106 :185,231,139,157,067,003,024
36112 :185,232,139,157,066,003,030
36118 :056,169,180,229,106,168,162
36124 :185,231,139,157,070,003,045
36130 :185,232,139,157,069,003,051
36136 :096,133,089,132,090,169,237
36142 :000,133,091,133,092,162,145
36148 :017,024,102,092,102,091,224
36154 :102,090,102,089,144,013,086
36160 :024,165,087,101,091,133,153
36166 :091,165,088,101,092,133,228
36172 :092,202,208,230,096,032,168
36178 :041,141,006,090,038,091,233
36184 :038,092,006,090,038,091,187
36190 :038,092,165,091,164,092,224
36196 :096,173,084,003,172,085,201
36202 :003,162,000,032,189,140,120
36208 :173,086,003,172,087,003,124
36214 :162,006,032,189,140,173,052
36220 :088,003,172,089,003,162,129
36226 :012,032,189,140,173,083,247
36232 :003,077,068,003,141,105,021
36238 :003,173,081,003,133,087,110
36244 :173,082,003,133,088,173,032
36250 :066,003,172,067,003,032,241
36256 :081,141,141,103,003,140,001
36262 :104,003,173,083,003,077,097
36268 :071,003,141,108,003,173,159
36274 :069,003,172,070,003,032,015
36280 :081,141,141,106,003,140,028
36286 :107,003,173,083,003,077,124
36292 :077,003,073,001,141,120,099
36298 :003,173,075,003,172,076,192
36304 :003,032,081,141,141,118,212
36310 :003,140,119,003,173,074,214
36316 :003,077,108,003,141,114,154
36322 :003,173,072,003,133,087,185
36328 :173,073,003,133,088,173,107
36334 :106,003,172,107,003,032,149
36340 :081,141,174,114,003,208,197
36346 :008,141,112,003,140,113,255
36352 :003,240,015,056,169,000,227
36358 :229,091,141,112,003,169,239
36364 :000,229,092,141,113,003,078
36370 :173,074,003,077,105,003,197
36376 :073,001,141,117,003,173,020
36382 :103,003,172,104,003,032,191
36388 :081,141,174,117,003,208,248
36394 :008,141,115,003,140,116,053
36400 :003,240,015,056,169,000,019
36406 :229,091,141,115,003,169,034
36412 :000,229,092,141,116,003,129
36418 :174,105,003,240,017,056,154
36424 :169,000,237,103,003,141,213
36430 :103,003,169,000,237,104,182
36436 :003,141,104,003,174,108,105
36442 :003,240,017,056,169,000,063
36448 :237,106,003,141,106,003,180
36454 :169,000,237,107,003,141,247
36460 :107,003,173,080,003,077,039
36466 :068,003,073,001,141,114,002
36472 :003,173,078,003,133,087,085
36478 :173,079,003,133,088,173,007
36484 :066,003,172,067,003,032,219
36490 :081,141,174,114,003,208,091
36496 :021,024,109,112,003,141,042
36502 :112,003,152,109,113,003,208
36508 :141,113,003,016,046,238,201
36514 :114,003,016,024,056,173,036
36520 :112,003,229,091,141,112,088
36526 :003,173,113,003,229,092,092
36532 :141,113,003,048,005,206,184
36538 :114,003,016,017,056,169,049
36544 :000,237,112,003,141,112,029
36550 :003,169,000,237,113,003,211
36556 :141,113,003,173,080,003,205
36562 :077,071,003,073,001,141,164
36568 :117,003,173,069,003,172,241
36574 :070,003,032,081,141,174,138
36580 :117,003,208,021,024,109,198
36586 :115,003,141,115,003,152,251
36592 :109,116,003,141,116,003,216
36598 :016,046,238,117,003,016,170
36604 :024,056,173,115,003,229,084
36610 :091,141,115,003,173,116,129
36616 :003,229,092,141,116,003,080
36622 :048,005,206,117,003,016,153
36628 :017,056,169,000,237,115,186
36634 :003,141,115,003,169,000,201
36640 :237,116,003,141,116,003,136
36646 :173,080,003,077,077,003,195
36652 :073,001,141,111,003,173,034
36658 :075,003,172,076,003,032,155
36664 :081,141,141,109,003,140,159
36670 :110,003,173,072,003,173,083
36676 :073,003,032,081,141,133,019
36682 :087,132,088,173,080,003,125
36688 :077,074,003,077,071,003,129
36694 :141,105,003,173,069,003,060
36700 :172,070,003,032,081,141,079
36706 :174,105,003,208,021,024,121
36712 :109,103,003,141,103,003,054
36718 :152,109,104,003,141,104,211
36724 :003,016,046,238,105,003,015
36730 :016,024,056,173,103,003,241
36736 :229,091,141,103,003,173,100
36742 :104,003,229,092,141,104,039
36748 :003,048,005,206,105,003,254
36754 :016,017,056,169,000,237,129
36760 :103,003,141,103,003,169,162
36766 :000,237,104,003,141,104,235
36772 :003,173,080,003,077,074,062
36778 :003,077,068,003,073,001,139
36784 :141,108,003,173,066,003,158
36790 :172,067,003,032,081,141,166
36796 :174,108,003,208,021,024,214
36802 :109,106,003,141,106,003,150
36808 :152,109,107,003,141,107,051
36814 :003,016,046,238,108,003,108
36820 :016,024,056,173,106,003,078
36826 :229,091,141,106,003,173,193
36832 :107,003,229,092,141,107,135
36838 :003,048,005,206,108,003,091
36844 :016,017,056,169,000,237,219
36850 :106,003,141,106,003,169,002
36856 :000,237,107,003,141,107,075
36862 :003,096,173,092,003,077,186
36868 :105,003,133,105,173,103,114
36874 :003,133,087,173,104,003,001
36880 :133,088,173,090,003,172,163
36886 :091,003,032,041,141,165,239
36892 :091,164,092,166,105,208,086
36898 :006,133,038,132,039,240,110
36904 :013,056,169,000,229,091,086
36910 :133,038,169,000,229,092,195
36916 :133,039,173,095,003,077,060
36922 :108,003,133,105,173,106,174
36928 :003,133,087,173,107,003,058
36934 :133,088,173,093,003,172,220
36940 :094,003,032,041,141,165,040
36946 :091,164,092,166,105,208,140
36952 :013,024,101,038,133,038,179
36958 :152,101,039,133,039,076,122
36964 :115,144,056,165,038,229,079
36970 :091,133,038,165,039,229,033
36976 :092,133,039,173,098,003,138
36982 :077,111,003,133,105,173,208
36988 :109,003,133,087,173,110,227
36994 :003,133,088,173,096,003,114
37000 :172,097,003,032,041,141,110
37006 :165,091,164,092,166,105,157
37012 :208,016,024,101,038,133,156
37018 :038,152,101,039,133,039,144
37024 :016,036,230,105,016,019,070
37030 :056,165,038,229,091,133,110
37036 :038,165,039,229,092,133,100
37042 :039,048,004,198,105,016,076
37048 :013,056,169,000,229,038,177
37054 :133,038,169,000,229,039,030
37060 :133,039,165,038,133,087,023
37066 :165,039,133,088,173,101,133
37072 :003,160,000,032,041,141,073
37078 :165,090,133,087,165,091,177
37084 :133,088,173,102,003,160,111
37090 :000,032,041,141,165,105,198
37096 :208,011,024,173,099,003,238
37102 :101,090,133,038,076,253,161
37108 :144,056,173,099,003,229,180
37114 :090,133,038,173,092,003,011
37120 :077,114,003,133,105,173,093
37126 :112,003,133,087,173,113,115
37132 :003,133,088,173,090,003,246
37138 :172,091,003,032,041,141,242
37144 :165,091,164,092,166,105,039
37150 :208,006,133,040,132,041,078
37156 :240,013,056,169,000,229,231
37162 :091,133,040,169,000,229,192
37168 :092,133,041,173,095,003,073
37174 :077,117,003,133,105,173,150
37180 :115,003,133,087,173,116,175
37186 :003,133,088,173,093,003,047
37192 :172,094,003,032,041,141,043
37198 :165,091,164,092,166,105,093
37204 :208,013,024,101,040,133,091
37210 :040,152,101,041,133,041,086
37216 :076,112,145,056,165,040,178
37222 :229,091,133,040,165,041,033
37228 :229,092,133,041,173,098,106
37234 :003,077,120,003,133,105,043
37240 :173,118,003,133,087,173,039
37246 :119,003,133,088,173,096,226
37252 :003,172,097,003,032,041,224
37258 :141,165,091,164,092,166,189
37264 :105,208,016,024,101,040,126
37270 :133,040,152,101,041,133,238
37276 :041,016,036,230,105,016,088
37282 :019,056,165,040,229,091,250
37288 :133,040,165,041,229,092,100
37294 :133,041,048,004,198,105,191
37300 :016,013,056,169,000,229,151
37306 :040,133,040,169,000,229,029
37312 :041,133,041,165,040,133,233
37318 :087,165,041,133,088,173,117
37324 :101,003,160,000,032,041,029
```

```
37330 :141,165,105,208,011,024,096
37336 :173,100,003,101,090,133,048
37342 :040,076,234,145,056,173,178
37348 :100,003,229,090,133,040,055
37354 :169,159,197,038,176,002,207
37360 :133,038,169,199,197,040,248
37366 :176,002,133,040,096,173,098
37372 :084,003,172,085,003,162,249
37378 :000,032,189,140,173,086,110
37384 :003,172,087,003,162,006,185
37390 :032,189,140,173,088,003,127
37396 :172,089,003,162,012,032,234
37402 :189,140,173,077,003,077,173
37408 :080,003,141,126,003,173,046
37414 :075,003,133,087,173,076,073
37420 :003,133,088,173,078,003,010
37426 :172,079,003,032,081,141,046
37432 :141,124,003,140,125,003,080
37438 :173,077,003,077,068,003,207
37444 :141,129,003,173,066,003,071
37450 :172,067,003,032,081,141,058
37456 :141,127,003,140,128,003,110
37462 :173,077,003,077,083,003,246
37468 :141,138,003,173,081,003,119
37474 :172,082,003,032,081,141,097
37480 :141,136,003,140,137,003,152
37486 :173,074,003,073,001,141,063
37492 :132,003,173,072,003,141,128
37498 :130,003,173,073,003,141,133
37504 :131,003,173,071,003,077,074
37510 :080,003,073,001,141,135,055
37516 :003,173,069,003,133,087,096
37522 :173,070,003,133,088,173,018
37528 :078,003,172,079,003,032,007
37534 :081,141,141,133,003,140,029
37540 :134,003,173,071,003,077,113
37546 :083,003,141,123,003,173,184
37552 :081,003,172,082,003,032,037
37558 :081,141,174,123,003,208,144
37564 :008,141,121,003,140,122,211
37570 :003,240,015,056,169,000,165
37576 :229,091,141,121,003,169,186
37582 :000,229,092,141,122,003,025
37588 :173,068,003,077,074,003,098
37594 :077,080,003,141,123,003,133
37600 :173,066,003,133,087,173,091
37606 :067,003,133,088,173,072,254
37612 :003,172,073,003,032,081,088
37618 :141,133,087,132,088,173,228
37624 :078,003,172,079,003,032,103
37630 :081,141,174,123,003,208,216
37636 :021,024,109,121,003,141,167
37642 :121,003,152,109,122,003,008
37648 :141,122,003,016,046,238,070
37654 :123,003,016,024,056,173,161
37660 :121,003,229,091,141,121,222
37666 :003,173,122,003,229,092,144
37672 :141,122,003,048,005,206,053
37678 :123,003,016,017,056,169,174
37684 :000,237,121,003,141,121,163
37690 :003,169,000,237,122,003,080
37696 :141,122,003,169,000,133,120
37702 :106,032,002,148,173,139,158
37708 :003,141,147,003,173,140,171
37714 :003,141,148,003,173,141,179
37720 :003,141,149,003,173,142,187
37726 :003,141,150,003,173,143,195
37732 :003,141,151,003,173,144,203
37738 :003,141,152,003,173,145,211
37744 :003,141,153,003,173,146,219
37750 :003,141,154,003,024,165,096
37756 :106,105,010,133,106,032,104
37762 :002,148,173,147,003,133,224
37768 :251,173,148,003,133,252,072
37774 :173,139,003,141,147,003,236
37780 :133,253,173,140,003,141,223
37786 :148,003,133,254,032,242,198
37792 :138,173,149,003,133,251,239
37798 :173,150,003,133,252,173,026
37804 :141,003,141,149,003,133,230
37810 :253,173,142,003,141,150,016
37816 :003,133,254,032,242,138,218
37822 :173,151,003,133,251,173,050
37828 :152,003,133,252,173,143,028
37834 :003,141,151,003,133,253,118
37840 :173,144,003,141,152,003,056
37846 :133,254,032,242,138,173,162
37852 :153,003,133,251,173,154,063
37858 :003,133,252,173,145,003,167
37864 :141,153,003,133,253,173,064
37870 :146,003,141,154,003,133,050
37876 :254,032,242,138,165,106,157
37882 :201,090,240,003,076,122,214
37888 :147,096,056,169,090,229,019
37894 :106,032,066,151,173,123,145
37900 :003,141,068,003,173,121,009
37906 :003,172,122,003,032,081,175
37912 :141,141,066,003,140,067,070
37918 :003,173,129,003,141,074,041
37924 :003,173,127,003,172,128,130
37930 :003,032,081,141,141,072,000
37936 :003,140,073,003,173,135,063
37942 :003,141,080,003,173,133,075
37948 :003,172,134,003,032,081,229
37954 :141,141,078,003,140,079,136
37960 :003,165,106,032,066,151,083
37966 :173,126,003,141,071,003,083
37972 :173,124,003,172,125,003,172
37978 :032,081,141,141,069,003,045
37984 :140,070,003,173,132,003,105
37990 :141,077,003,173,130,003,117
37996 :172,131,003,032,081,141,156
38002 :141,075,003,140,076,003,040
38008 :173,138,003,141,083,003,149
38014 :173,136,003,172,137,003,238
38020 :032,081,141,141,081,003,099
38026 :140,082,003,032,010,149,042
38032 :165,038,141,139,003,165,027
38038 :040,141,140,003,173,071,206
38044 :003,073,001,141,071,003,192
38050 :173,077,003,073,001,141,118
38056 :077,003,173,083,003,073,068
38062 :001,141,083,003,032,010,188
38068 :149,165,038,141,141,003,049
38074 :165,040,141,142,003,173,082
38080 :068,003,073,001,141,068,034
38086 :003,173,074,003,073,001,013
38092 :141,074,003,173,080,003,166
38098 :073,001,141,080,003,032,028
38104 :010,149,165,038,141,145,096
38110 :003,165,040,141,146,003,208
38116 :173,071,003,073,001,141,178
38122 :071,003,173,077,003,073,122
38128 :001,141,077,003,173,083,206
38134 :003,073,001,141,083,003,038
38140 :032,010,149,165,038,141,019
38146 :143,003,165,040,141,144,126
38152 :003,096,173,066,003,141,234
38158 :090,003,173,067,003,141,235
38164 :091,003,173,068,003,240,086
38170 :017,056,169,000,237,090,083
38176 :003,141,090,003,169,000,182
38182 :237,091,003,141,091,003,092
38188 :173,071,003,141,092,003,015
38194 :208,026,024,173,069,003,041
38200 :109,090,003,141,090,003,236
38206 :173,070,003,109,091,003,255
38212 :141,091,003,016,048,238,093
38218 :092,003,016,026,056,173,184
38224 :090,003,237,069,003,141,111
38230 :090,003,173,091,003,237,171
38236 :070,003,141,091,003,048,192
38242 :005,206,092,003,016,017,181
38248 :056,169,000,237,090,003,147
38254 :141,090,003,169,000,237,238
38260 :091,003,141,091,003,173,106
38266 :072,003,141,093,003,173,095
38272 :073,003,141,094,003,173,103
38278 :074,003,240,017,056,169,181
38284 :000,237,093,003,141,093,195
38290 :003,169,000,237,094,003,140
38296 :141,094,003,173,077,003,131
38302 :141,095,003,208,026,024,143
38308 :173,075,003,109,093,003,108
38314 :141,093,003,173,076,003,147
38320 :109,094,003,141,094,003,108
38326 :016,048,238,095,003,016,086
38332 :026,056,173,093,003,237,008
38338 :075,003,141,093,003,173,170
38344 :094,003,237,076,003,141,242
38350 :094,003,048,005,206,095,145
38356 :003,016,017,056,169,000,217
38362 :237,093,003,141,093,003,020
38368 :169,000,237,094,003,141,100
38374 :094,003,173,078,003,141,210
38380 :096,003,173,079,003,141,219
38386 :097,003,173,080,003,240,070
38392 :017,056,169,000,237,096,055
38398 :003,141,096,003,169,000,154
38404 :237,097,003,141,097,003,070
38410 :173,083,003,141,098,003,255
38416 :208,026,024,173,081,003,019
38422 :109,096,003,141,096,003,214
38428 :173,082,003,109,097,003,239
38434 :141,097,003,016,048,238,065
38440 :098,003,016,026,056,173,156
38446 :096,003,237,081,003,141,095
38452 :096,003,173,097,003,237,149
38458 :082,003,141,097,003,048,176
38464 :005,206,098,003,016,017,153
38470 :056,169,000,237,096,003,119
38476 :141,096,003,169,000,237,210
38482 :097,003,141,097,003,173,084
38488 :161,003,133,087,173,162,039
38494 :003,133,088,173,090,003,072
38500 :172,091,003,032,081,141,108
38506 :173,092,003,240,013,056,171
38512 :169,000,229,091,133,091,057
38518 :169,000,229,092,133,092,065
38524 :024,165,091,109,155,003,159
38530 :141,090,003,165,092,109,218
38536 :156,003,141,091,003,169,187
38542 :000,141,092,003,173,091,130
38548 :003,016,020,238,092,003,008
38554 :056,169,000,237,090,003,197
38560 :141,090,003,169,000,237,032
38566 :091,003,141,091,003,173,156
38572 :093,003,172,094,003,032,057
38578 :081,141,173,095,003,240,143
38584 :013,056,169,000,229,091,230
38590 :133,091,169,000,229,092,136
38596 :133,092,024,165,091,109,042
38602 :157,003,141,093,003,165,252
38608 :092,109,158,003,141,094,037
38614 :003,169,000,141,095,003,113
38620 :173,094,003,016,020,238,252
38626 :095,003,056,169,000,237,018
38632 :093,003,141,093,003,169,222
38638 :000,237,094,003,141,094,039
38644 :003,173,096,003,172,097,020
38650 :003,032,081,141,173,098,010
38656 :003,240,013,056,169,000,225
38662 :229,091,133,091,169,000,207
38668 :229,092,133,092,024,165,235
38674 :091,109,159,003,141,096,105
38680 :003,165,092,109,160,003,044
38686 :141,097,003,169,000,141,069
38692 :098,003,173,097,003,016,170
38698 :020,238,098,003,056,169,114
38704 :000,237,096,003,141,096,109
38710 :003,169,000,237,097,003,051
38716 :141,097,003,076,000,144,009
38722 :010,168,185,231,139,133,164
38728 :088,185,232,139,133,087,168
38734 :096,032,027,134,201,001,057
38740 :208,047,173,164,003,240,151
38746 :003,076,053,164,238,165,021
38752 :003,169,119,205,165,003,248
38758 :208,003,238,164,003,173,123
38764 :170,003,172,171,003,032,147
38770 :231,151,173,170,003,105,179
38776 :011,141,170,003,173,171,021
38782 :003,105,000,141,171,003,037
38788 :096,201,002,208,047,173,091
38794 :166,003,240,003,076,053,167
38800 :164,238,167,003,169,080,197
38806 :205,167,003,208,003,238,206
38812 :166,003,173,172,003,172,077
38818 :173,003,032,231,151,173,157
38824 :172,003,105,011,141,172,004
38830 :003,173,173,003,105,000,119
38836 :141,173,003,096,173,168,166
38842 :003,240,003,076,053,164,213
38848 :238,169,003,169,080,205,032
38854 :169,003,208,003,238,168,219
38860 :003,173,174,003,172,175,136
38866 :003,032,231,151,173,174,206
38872 :003,105,011,141,174,003,141
38878 :173,175,003,105,000,141,051
38884 :175,003,096,133,029,132,028
38890 :030,032,024,134,032,041,015
38896 :134,160,002,138,145,029,080
38902 :136,165,100,145,029,136,189
38908 :165,101,145,029,032,024,236
38914 :134,032,041,134,160,005,252
38920 :138,145,029,136,165,100,209
38926 :145,029,136,165,101,145,223
38932 :029,032,024,134,032,041,056
38938 :134,160,008,138,145,029,128
38944 :136,165,100,145,029,136,231
```

```
38950 :165,101,145,029,032,024,022
38956 :134,160,009,145,029,032,041
38962 :024,134,160,010,145,029,040
38968 :024,096,032,027,134,201,058
38974 :001,208,025,173,165,003,125
38980 :208,003,076,053,164,173,233
38986 :165,003,141,176,003,169,219
38992 :000,133,029,169,192,133,224
38998 :030,076,141,152,201,002,176
39004 :208,025,173,167,003,208,108
39010 :003,076,053,164,173,167,222
39016 :003,141,176,003,169,029,113
39022 :133,029,169,197,133,030,033
39028 :076,141,152,173,169,003,062
39034 :208,003,076,053,164,173,031
39040 :169,003,141,176,003,169,021
39046 :141,133,029,169,200,133,171
39052 :030,160,000,177,029,141,165
39058 :090,003,200,177,029,141,018
39064 :091,003,200,177,029,141,025
39070 :092,003,200,177,029,141,032
39076 :093,003,200,177,029,141,039
39082 :094,003,200,177,029,141,046
39088 :095,003,200,177,029,141,053
39094 :096,003,200,177,029,141,060
39100 :097,003,200,177,029,141,067
39106 :098,003,200,177,029,141,074
39112 :065,003,200,177,029,208,114
39118 :021,032,000,144,165,038,094
39124 :133,251,133,025,165,040,191
39130 :133,252,133,026,032,108,134
39136 :138,076,242,152,032,000,096
39142 :144,165,038,133,253,165,104
39148 :040,133,254,032,242,138,051
39154 :206,176,003,240,016,024,139
39160 :165,029,105,011,133,029,208
39166 :165,030,105,000,133,030,205
39172 :076,141,152,096,032,027,016
39178 :134,201,001,208,017,169,228
39184 :000,141,164,003,141,165,118
39190 :003,141,170,003,169,192,188
39196 :141,171,003,096,201,002,130
39202 :208,019,169,000,141,166,225
39208 :003,141,167,003,169,029,040
39214 :141,172,003,169,197,141,101
39220 :173,003,096,169,000,141,122
39226 :168,003,141,169,003,169,199
39232 :141,141,174,003,169,200,124
39238 :141,175,003,096,173,003,149
39244 :220,240,027,165,198,208,110
39250 :020,164,190,177,195,240,044
39256 :009,141,119,002,230,198,019
39262 :230,190,208,005,169,000,128
39268 :141,003,220,076,066,235,073
39274 :173,141,002,208,035,165,062
39280 :203,201,004,208,007,169,136
39286 :183,160,153,076,149,153,224
39292 :201,005,208,007,169,193,139
39298 :160,153,076,149,153,201,254
39304 :006,208,007,169,203,160,121
39310 :153,076,149,153,076,072,053
39316 :235,133,195,132,196,165,180
39322 :198,208,023,032,072,235,154
39328 :165,198,240,016,198,198,151
39334 :169,016,141,003,220,169,116
39340 :000,141,001,220,169,000,191
39346 :133,190,076,066,235,003,193
39352 :067,082,069,069,078,032,069
39358 :048,013,000,083,067,082,227
39364 :069,069,078,032,049,013,250
39370 :000,083,067,082,069,069,060
39376 :078,032,050,013,000,013,138
```

Program 2: Complex Animation Demo

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
10 REM SET FIGURE 1          :rem 36
20 FCLEAR 1                  :rem 64
30 FOR I=1 TO 33 : READ X,Y,Z,
   CO,A                      :rem 238
40 FSET 1,X*500,Y*500,Z*500,CO
   ,A                        :rem 190
50 NEXT I                    :rem 237
60 REM SET FIGURE 2          :rem 42
70 FCLEAR 2 :RESTORE         :rem 164
80 FOR I=1 TO 33 : READ X,Y,Z,
   CO,A                      :rem 243
90 FSET 2, Y*500, X*500,-Z*500
   ,4-CO,A                   :rem 82
100 NEXT I                   :rem 25
110 REM{3 SPACES}ANIMATION OF
    {SPACE}THE CROSS         :rem 22
120 A=1:B=2:C=3              :rem 26
130 PARAM 80,100,25,165
                             :rem 133
140 COLOR 12,11,10,13,14
                             :rem 179
150 FORJ=10TO50STEP10        :rem 10
160 FORI=10TO80STEP10        :rem 13
170 ANGLEI,J,I{4 SPACES}:ANIM1
    :FIGURE1:FIGURE2         :rem 30
180 ANGLEI+5,J,I+5:ANIM2:FIGUR
    E1:FIGURE2               :rem 224
190 NEXTI,J                  :rem 152
200 FORI=1TO2000:NEXT        :rem 14
210 SCREEN0:REM NORMAL:rem 106
220 END                      :rem 107
230 REM CO-ORDINATES OF THE CR
    OSS                      :rem 225
240 DATA -1,-1,-1,1,0,-1,-4,-1
    ,1,1,-1,-4,-3,1,1 :rem 147
250 DATA -1,4,-3,1,1,-1,4,-1,1
    ,1,1,4,-1,1,1,1,4 :rem 161
260 DATA -3,1,1,1,-4,-3,1,1,1,
    -4,-1,1,1,1,-1,-1,1
                             :rem 248
270 DATA 1,-1,-4,-1,1,0,1,-4,-
    1,1,1,-1,-4,-3,1,0:rem 200
280 DATA 1,-4,-3,1,1,-1,4,-3,1
    ,0,1,4,-3,1,1,1,1 :rem 164
290 DATA -1,1,0,1,4,-1,1,1,-1,
    1,-1,1,0,-1,4,-1,1:rem 200
300 DATA 1,-1,-1,-1,2,0,1,-1,-
    1,2,1,1,-1,-2,2,1 :rem 143
310 DATA -1,-1,-2,2,1,-1,1,-2,
    2,1,-1,1,-1,2,1,1,1
                             :rem 239
320 DATA -1,2,1,1,1,-2,2,1,1,-
    1,-2,2,1,1,1,-2,2,0
                             :rem 244
330 DATA -1,1,-2,2,1,-1,-1,-1,
    2,0,-1,-1,-2,2,1  :rem 99
```

Program 3: Observation Angles Demo

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
10 PARAM 80,100,30,165 :rem 78
20 COLOR 12,11,10,13,14
                          :rem 128
30 FORJ=5TO355STEP10      :rem 227
40 FORI=0TO340STEP20      :rem 217
50 ANGLE J,I,I{6 SPACES}:ANIM2
   :GOSUB100              :rem 220
60 ANGLE J,I+10,I+10:ANIM1:GOS
   UB100                  :rem 244
70 NEXTI,J                :rem 101
80 FORI=1TO500:NEXT       :rem 183
90 SCREEN0:END            :rem 106
100 DLINE-500,0,0,2000,0,0,1
                          :rem 122
110 DLINE0,-500,0,0,2000,0,2
                          :rem 124
120 DLINE0,0,-500,0,0,2000,3
                          :rem 126
130 RETURN                :rem 116
```

©

IBM Graphics On A Monochrome Monitor

Thomas G. Hanlin III

Though advanced IBM graphics require a color/graphics adapter, you can create simple graphics and even animation on a monochrome system as well. Here's a short program to show how it's done.

IBM PC computers can generate stunning graphics, but advanced BASIC graphics features are available only on PCjrs or PCs with a color/graphics adapter. However, with the right programming methods, your monochrome system can produce graphics, too. Granted, they are fairly low resolution—and no amount of programming skill can make your monochrome monitor display more than one color—but they are graphics nonetheless. You may find them handy for utilitarian purposes (for example, adding interest to bar graph displays), or you may enjoy making simple graphic screens, animated figures, or games. Once you master the basic technique, more and more applications will come to mind.

Character Graphics

When an IBM PC boots up, it checks to see if the system contains

a color/graphics adapter and configures itself accordingly. If a color/graphics adapter is present, you may use advanced BASIC graphics commands like PUT and GET. If not, those commands cause an error. However, even a monochrome system has the ability to display a large set of special characters. IBM graphics characters have ASCII values of 128 to 255 and include a number of different shapes useful in creating boxes, borders, and so on.

The characters we're interested in are those which consist of a solid block. All computer graphics are produced by turning *pixels* (picture elements) on or off to light up different parts of the screen. The smaller the size of the pixel dots, the more detailed the image. Although the IBM character set doesn't include any pixel-sized characters—each character is composed of several pixels—it does include some we can use like giant pixels.

Giant Pixels

For example, CHR$(219) is a solid block character, the inverse of CHR$(32), the blank space. Using these two characters together provides a graphics screen with 80 × 25 resolution. To turn on a "dot" within this coarse screen, print the solid block at the desired spot. To turn off a dot, print a space. The BASIC function SCREEN(Y,X) tells you whether a given location contains a dot or an empty space. Though you're limited to simple, quite blocky shapes, this system is fast and simple to use. However, it's possible to do much better.

Besides the block and space characters which light up or blank out an entire screen location, there are some which light up only part of a screen position. For instance, CHR$(220) is solid on the bottom half and blank on the top. The reverse is true of CHR$(223). By using these characters, we can double our resolution to 80 × 50 pixels. This complicates matters a bit, since we want to use only half a screen position, and BASIC lets you print only to an entire screen position. Here's a point-plotting routine that handles the tricky details for you:

```
10000 GR.Y=Y\2+1:GR.SC=SCREEN
      (GR.Y,X+1):GR.OFFSET=(Y
       MOD 2)*3:IF Z=0 THEN 1
      0020 ELSE IF GR.SC=32 T
      HEN GR.SC=223-GR.OFFSET
       ELSE IF GR.SC+GR.OFFSE
      T<>223 THEN GR.SC=219
10010 LOCATE GR.Y,X+1:PRINT C
      HR$(GR.SC);:RETURN
10020 IF GR.SC+GR.OFFSET=223
      THEN GR.SC=32 ELSE IF G
      R.SC<>32 THEN GR.SC=220
      +GR.OFFSET
10030 GOTO 10010
10040 GR.Y=Y\2+1:S9=SCREEN(GR
      .Y,X+1):Z=(GR.SC=219 OR
       GR.SC+(Y MOD 2)*3=223)
      :RETURN
```

To plot a point with this routine, set the variable X to the desired horizontal coordinate (0–79) and the variable Y to the vertical coordinate (0–49). Now you've set the screen location for the giant pixel. To turn it on, set the variable Z to 1. Set Z to 0 to turn the pixel off. Then call the subroutine with GOSUB 10000. Line 10040 is a separate routine that tells you whether a given location is lit up or blank. To test any point on the screen, set the variables X and Y to the appropriate coordinates; then GOSUB 10040. The variable Z equals −1 if that point is lit or 0 if it's blank.

An Animated Snake

Though this system emulates a simple graphics screen, keep in mind that you are still printing characters. Thus, there are four screen locations that cause everything to scroll upward if you plot a point there: locations (79,46), (79,47), (79,48), and (79,49). To avoid scrolling your display, either do not use these particular locations or restrict your screen to 79 × 50 pixels (use horizontal locations 0–78). Note that you can mix text and graphics freely, but putting graphics on top of text causes some surprising results. The following program demonstrates how to animate a simple figure. Add these lines to the point-plotting routine and save the program. Make sure the numeric keypad is in numeric mode before you run it.

```
10 KEY OFF:CLS:DEFINT A-Z:Y=0
   :Z=1:FOR X=0 TO 24:SNAKE$=
   BNAKE$+CHR$(X)+CHR$(Y):GOS
   UB 10000:NEXT:DX=1:DY=0:X=
   X-1
20 I$=INKEY$:IF I$<>"" THEN D
   X=SGN(INSTR("369",I$)-INST
   R("147",I$)):DY=SGN(INSTR(
   "123",I$)-INSTR("789",I$))
   :IF I$=" " THEN CLS:END
30 X=ASC(RIGHT$(SNAKE$,2))+DX
   :Y=ASC(RIGHT$(SNAKE$,1))+D
   Y:IF X>78 THEN X=0 ELSE IF
    X<0 THEN X=78
40 IF Y>49 THEN Y=0 ELSE IF Y
   <0 THEN Y=49
50 Z=1:GOSUB 10000:SNAKE$=SNA
   KE$+CHR$(X)+CHR$(Y):X=ASC(
   LEFT$(SNAKE$,1)):Y=ASC(MID
   $(SNAKE$,2,1)):Z=0:GOSUB 1
   0000:SNAKE$=MID$(SNAKE$,3)
   :GOTO 20
```

Control the direction of the wandering animated snake by using the numeric keypad. Press the space bar to end the program. To improve its speed, the point-plotting routine is as short as possible. However, if you don't require fast drawing, you might want to add other features. Perhaps you'd like to color or shade the points to introduce different degrees of brightness (of course, since each two-pixel pair corresponds to a single character, there's a limit to this technique). You might add range checking to check for valid coordinates before you plot a point. And you could also modify the routine to place graphics on top of text correctly. ©

COMMODORE
Dynamic Keyboard
Part 2

Jim Butterfield, Associate Editor

Part 1 discussed the fundamentals of dynamic keyboard programming, which in effect allows the computer to "type on its own keyboard." Now let's look at some important applications for this technique.

As we saw in Part 1, dynamic keyboard programming uses a two-step method to let a program give itself direct-mode commands. Step 1 is to print the command at a specific location on the screen. Step 2 is to put a RETURN character in the computer's keyboard buffer, then stop the program with the cursor flashing over the screen command. The RETURN character makes the computer execute the command just as if you'd pressed RETURN.

It's worth mentioning that you may print more than one command on a screen line. Just as in a program line, separate the multiple direct-mode commands with colons. You can use more than one screen line of direct-mode commands as well. However, you must be careful to put the commands in exactly the right place, and make sure the cursor flashes directly over the line to be executed when the program stops.

Here are some applications for the dynamic keyboard technique:

- Allow a user to enter a formula that the program will use;
- Allow a program to load another program;
- Allow a program to modify itself (tricky);
- Run test programs to determine, for instance, how the computer responds to certain direct commands and calculations.

Keyboard Buffer Locations

The following table shows the location of the keyboard buffer counter and the start of the keyboard buffer on most Commodore computers:

	Counter	Buffer
VIC-20, Commodore 64	198	631
Commodore 16, Plus/4	239	1319
PET/CBM (4.0 & Upgrade BASIC)	158	623
PET (Original ROM)	525	527
B128 (Model 700)	209	939

Usually your program must POKE a value of 1 into the counter and a value of 13 (the character code for RETURN) into the buffer. That tells the computer there's one RETURN character in the buffer waiting to be processed. If there's more than one line of direct-mode commands on the screen to be performed, you need a higher count and more characters. On the B128, it's wise to execute a BANK 15 command before the POKEs.

Entering A Formula

Let's write a brief program that allows a student to enter a formula and then generates a table of values based on the formula. More complex versions of the program might solve an equation or draw a graph, but we'll keep the example simple. In practice, it would be wise for your program to check for valid syntax before evaluating the formula. Again, for the sake of brevity, we'll do only the dynamic keyboard portion.

This program is for VIC-20 and Commodore 64 only. If you have another Commodore model, use the table above to change the POKE addresses in line 140. Also, don't forget the colon that appears just before the GOTO statement in line 130.

```
100 PRINT"{CLR}{DOWN}FORMULA E
    VALUATION.":PRINT"INPUT A
    {SPACE}FORMULA"      :rem 52
110 PRINT"BASED ON VARIABLE X"
    :PRINT"SUCH AS:":PRINT"
    {DOWN}{2 SPACES}Y= X*7-SQR
    (X)":PRINT              :rem 7
120 PRINT"YOUR FORMULA:":INPUT
    "{DOWN}{2 SPACES}Y=";F$:PR
    INTCHR$(147):PRINT:PRINT
                          :rem 160
130 PRINT"Y=";F$;":GOTO150":DI
    MV(10):FORX=1TO10:PRINTCHR
    $(19)                 :rem 178
140 POKE 198,1:POKE631,13:END
                          :rem 103
150 V(X)=Y:NEXT X:FOR X=1 TO 1
    0:PRINT X,V(X):NEXT X
                            :rem 2
```

Notice how this program does a task which would be difficult or impossible without using the dynamic keyboard technique.

Loading Another Program

If you put a LOAD command in a program, the new program doesn't load in the usual way. Instead, it's *chained* to the old program. The new program retains the variables and arrays (if certain rules are observed), and the effect is that of two successive programs working continuously on a single job. That's not always what is wanted. Especially with menu programs or bootstraps (program-loading programs), your goal may be simply to start the new program without preserving

variables or data from the old one. That's what happens when you perform LOAD as a direct command. With the dynamic keyboard technique, we can simulate this from within a program.

Let's write a simple dynamic keyboard loading sequence. Again, the program is given for VIC-20 and Commodore 64 only. For other Commodore models, use the table above to change the POKE addresses in line 120.

```
100 PRINT"{CLR}{DOWN}PROGRAM L
    OADING":PRINT"PROGRAM
    {2 DOWN}":PRINT"PROGRAM NA
    ME":INPUTP$
110 PRINT"{CLR}":PRINT:PRINT:P
    RINT"LOAD";CHR$(34);P$;CHR
    $(34);",8":PRINT:PRINT
120 PRINT:PRINT:PRINT"RUN":PRI
    NTCHR$(19):POKE198,2:POKE6
    31,13:POKE632,13
```

Note that there are two separate command lines: one for LOAD and one for RUN. Of course, it's important to position the lines correctly, but that's not hard to work out when you set up the program. You see everything happening on the screen, and, if you've placed your command a line too high or low, the problem is easy to spot. (For the VIC, you must limit the length of the filename you enter to seven or fewer characters. Otherwise, an unrelated bug built into the VIC's INPUT statement causes the program to fail.)

Tricks And Advanced Points

On computers with color capabilities, you can hide your dynamic keyboard tricks if you wish. If you print the direct-mode commands in the same character color as the screen background, they won't be visible to you, but the computer can still see and execute them. Your program can even change colors as it runs so that some parts of the commands are visible and some are not.

Occasionally, you'll want to use the dynamic keyboard technique to change a program as it runs. That's tricky, since any time you add or change a program line, the values of all variables are lost. It's hard to run a program when its variables disappear, but it can be done if handled carefully. The critical variables can be reentered using the dynamic keyboard technique, using lines such as X=7:L=120: GOTO 580. Another, somewhat more cumbersome method is to POKE the value of each variable into spare memory and PEEK the value later when needed.

Why would a program need to change itself? The most usual situation involves converting an ASCII program listing into tokenized BASIC format. It's common to list programs in ASCII (untokenized) form when translating from one computer to another. This is especially true when you transfer programs over the phone line with a modem. As each line of the ASCII listing arrives, it must be entered as if it were being typed, to store it in tokenized format. While it's possible to do the whole job by hand (by printing each line on the screen and pressing RETURN), the dynamic keyboard technique lets the computer do this busywork for you.

Next month, in Part 3, we'll cover the use of the dynamic keyboard technique for self-modifying programs in more detail. ©

Simple Assembling With IBM DEBUG

Tim Victor, Editorial Programmer

You don't need to buy an expensive assembler to write short machine language programs on an IBM PC or PCjr—a copy of PC-DOS already contains the basic tools you require. This article, which assumes some familiarity with hexadecimal numbers and machine language theory, shows how to make the most of the DEBUG *utility when you're ready to tackle 8088 machine language.*

Tucked away on the *DOS Supplemental Programs* disk that came with your copy of PC-DOS is a file called DEBUG. DEBUG is a simple but powerful development tool for exploring your computer and writing short machine language (ML) programs. It includes a *miniassembler,* which converts assembly language instructions into machine language directly in memory, and a *disassembler,* which allows you to reverse this process and examine ML programs already in memory. DEBUG also has trace and breakpoint functions for testing ML programs, utilities for loading and saving programs on disk, and several other valuable features. Using these tools, we'll show how to write a small ML program.

To get started with DEBUG, boot up DOS from your master disk. When the DOS prompt A> appears, insert the *DOS Supplemental Programs* disk into drive A:, type DEBUG, and press ENTER. DEBUG loads and runs, replacing the DOS prompt with its own prompt, a hyphen (-). You can return to DOS at any time by putting your master disk back in the drive, typing Q for Quit, and pressing ENTER.

Since you should preserve your *DOS Supplemental Programs* disk as an archival backup, let's ask DEBUG to copy itself onto another disk. You could use the DOS COPY command, but using DEBUG is a good way to learn how to load and save machine language program files.

Cloning DEBUG

DEBUG has three commands for disk operations: L (Load), W (Write), and N (Name). N creates a data structure called a *file control block* (FCB) that DOS uses for all disk operations, including DEBUG's Load and Write. The FCB contains the name of a file, along with information such as size and file organization. To learn more about the FCB, consult Appendix E of the *DOS 2.00 Manual,* or Chapter 6 of the *DOS 2.10 Technical Reference Manual.*

The first step in backing up DEBUG is to load another copy of it into memory. Type N DEBUG.COM and press ENTER. (You need to include the .COM extension because DEBUG doesn't make any assumptions about the file type.) DEBUG responds with another hyphen. Next, type L and press ENTER. The disk drive whirs, and then another hyphen appears. You've loaded a second copy of DEBUG.

Remove the *Supplemental Programs* disk. Replace it with a formatted disk that you'll be using for ML programs. Type W and press ENTER. The drive comes on again, and then DEBUG displays the message "Writing 2E80 bytes" and the hyphen prompt. You now have a copy of DEBUG.COM on your ML disk.

A Sample Program

Let's try assembling a program with DEBUG. Start by typing A 100 to start assembling at address 100H. (IBM programmers generally denote hexadecimal numbers by appending an H to the number. All input and output with DEBUG is expressed in hexadecimal.) DEBUG responds with *xxxx*:0100, where *xxxx* is a four-digit hexadecimal number. This number is the current value of the *code segment register.* It's of minor importance right now and will be discussed in detail later.

Now type in the following program. DEBUG displays the memory address of each instruction for you. All you need to enter are the instructions.

```
MOV AH,09
MOV DX,109
INT 21
INT 20
DB "HELLO THERE$"
```

Press ENTER to leave the assembler. This program is the ML equivalent of everyone's first BASIC program:

```
10 PRINT "HELLO THERE"
```

The ML version looks quite a bit longer, but it would be even more involved if it weren't for the INT 21H instruction, which calls a DOS function routine (Print String) by executing a software INTerrupt. Before calling this routine, the program takes two preparatory actions. The first instruction loads the AH register (an internal 8088 register) with the value 9. In 8088 machine language, instructions with two operands like MOV AH,09 operate from right to left—just as A=9 in

BASIC *moves* the value 9 into the variable A. You specify the destination operand first, then the source operand. This might seem a little backwards, but it's a common convention and you'll soon adjust to it.

AH is the high (most significant) byte of AX, the 16-bit (two-byte) accumulator register of the 8088. When a program calls Interrupt 21H, the value in AH indicates the function you're asking DOS to perform. Function 9, Print String, displays a string on the screen, starting with the character at the address contained in the DX register and ending with the character $. The second instruction moves the address 109H into the DX register. The last instruction, INT 20H, ends the program by returning control to the program that called it—in this case, DEBUG.

Finally, we create the string we want to print using DB, a pseudo-opcode (*pseudo-op*). When the assembler sees a pseudo-op such as DB, it performs a function instead of generating code. This particular pseudo-op tells the assembler to store bytes of data in memory, beginning at the current location. The data can be either a list of hexadecimal numbers between 00 and FF, separated by spaces or commas, or a quoted string, as shown above. If the data is a string, the ASCII code for each character is entered in memory. The dollar sign at the end of the string is very important. Without this *delimiter*, the Print String function will keep printing whatever bytes it happens to find in memory following the message. It might be a long time before it comes across a $ and stops.

8088 Memory Addressing

Now that the program is in memory, we can use the disassembler to examine it. Type U for Unassemble, and DEBUG displays several rows of text on the screen (the number of rows differs between 40- and 80-column displays). Notice that the disassembled code is aligned in four columns. The first column shows the address of each instruction as two four-digit hexadecimal numbers separated by a colon, just as was displayed when you entered the program. The first four-digit number is the current value of the code segment register mentioned before, and the second is the value of the *instruction pointer*. To understand why two registers are needed to point to a single memory location requires some understanding of the 8088's addressing scheme.

The 8088 microprocessor can access up to one megabyte (1024K) of memory using 20-bit addresses. However, for compatibility with older Intel processors, the 8088 has only a 16-bit instruction pointer. Because a 16-bit (four hexadecimal digit) register can only have values between 0 and 65,535, another register, the code segment register, is needed to address the entire 1,048,576 bytes allowed by the 8088. The code segment register is also a 16-bit register, but instead of addressing individual bytes, it points to blocks of 16 bytes, called *paragraphs*. Any five-digit hexadecimal address that ends in a zero is the beginning of a paragraph. For example, the byte of memory at 5D320H is at the beginning of the paragraph addressed by a segment register containing 5D32H.

The code segment register points to the first paragraph of a 64K block of memory called the *code segment* (CS). There are three other segments, the *data segment* (DS), *stack segment* (SS), and *extra segment* (ES), plus a register that points to the beginning of each. In simple programs, however, all the segment registers usually have the same value as CS. To find the next byte of code to be fetched, the value in the instruction pointer is added to the address of the beginning of the code segment. The physical address of this byte can be found with this formula:

Physical Address = IP+(CS*16)

The effect of organizing memory this way is that a programmer doesn't have to know where the program will be loaded. When DOS loads a .COM program, it starts the code segment at the beginning of any available paragraph in memory. The program is loaded at an offset of 100H bytes above the start of the segment and the instruction pointer is set to 100H. The four segment registers, CS, DS, SS, and ES, all point to the start of the code segment.

The second instruction of the example program moves an address, 109H, into DX. This address is an offset into the current data segment. The string to be printed is located at an offset of 109H only if the data segment register is equal to the code segment register and the program starts at offset 100H. In practice, the CS register is rarely changed except by DOS and needs little or no attention in most programs.

Displaying Binary Code

The second column of the disassembled listing on the screen contains four- or six-digit hexadecimal numbers. These are the contents of the memory locations, the binary code which the 8088 can execute. Notice that the first MOV instruction is one byte shorter than the second. The first instruction only loads half of a 16-bit register (AH is the upper half of AX), so the data occupies one byte, but the second MOV loads all of DX, which takes two bytes of data (a *word*).

The third column shows the *mnemonics*—symbolic names for each opcode instruction. The fourth column displays the operands. This program consists of four opcodes: two MOV instructions followed by two INT instructions. Notice that the DB pseudo-op doesn't show up in a disassembly. Instead of displaying your characters, DEBUG tries to convert the string into assembler mnemonics, and therefore prints several meaningless instructions. DEBUG is frequently fooled this way because program instructions and data are both stored as binary bytes. DEBUG has no way of knowing where the program ends and the data begins.

If you type another U, DEBUG continues to disassemble and display the next 16 or 32 bytes in memory (depending on your screen width). Since the program is only 21 bytes long, DEBUG starts displaying part of itself, still in memory from when you copied it. Type U 100 to disassemble from the beginning of your program again. DEBUG's U command also accepts both starting and ending addresses if you separate them with a space.

It's a good idea to save your program on disk before running it.

If the program causes something unexpected, like an infinite loop or a complete system crash, it's nice to have a copy saved. Then you can load it and search for the error without typing the program again from scratch.

As before, you need to tell DEBUG the name of your file. Type N HELLO.COM. Now there's one more thing to consider: How many bytes of memory should DEBUG write to disk? When we used the W command to copy DEBUG, it wrote the same number of bytes that it had loaded, but now we're saving a new program which has never been loaded. When DEBUG loads a file, it stores the size of the file in the CX register and the four least significant bits of the BX register. The same registers are used when DEBUG writes a file. So if your program is less than 65,536 bytes long (most are), the BX register should be set to zero.

To examine and change CX, type R CX. DEBUG prints the contents of CX (probably 2E80H, left over from copying DEBUG), then prints a colon at the beginning of the next line. You can press ENTER to leave the value unchanged, or type a new value. Since the new program is 21 bytes long, type 15 (the hexadecimal equivalent of 21) and press ENTER. Now type W to write the program to disk. DEBUG responds with the message "Writing 0015 bytes," then returns the prompt.

Running And Debugging

Now that your program is safe on disk, run it by typing G and pressing ENTER. The screen should display HELLO THERE. Then DEBUG prints "Program completed normally" followed by its usual prompt. If your program completed but didn't print correctly, disassemble starting from 100H and check that all instructions are correct. If your program locked up the computer, reboot, restart DEBUG, and thank yourself for saving the program. Reload the program with N and L, then disassemble it to see what it looks like. If you don't know what's wrong, one technique is to try setting a *breakpoint*. This halts the program at a predetermined point so you can check the contents of the registers.

For instance, to make the program stop before the INT 20H instruction, you can set one or more breakpoints. To set a breakpoint, type G followed by the addresses of one or more instructions in your program. If you set more than one breakpoint, separate the addresses with spaces. The program begins executing, but stops when the instruction pointer equals the address of a breakpoint. DEBUG displays the contents of all registers and flags and disassembles the instruction at the breakpoint (the instruction pointed to by the instruction pointer, the next instruction to be executed). Type G to restart the program at the instruction that the instruction pointer references.

If you stopped your program with a breakpoint but want to restart it from the beginning, type G =100. DEBUG sets the instruction pointer to 100H (or whatever address you specify) before starting. You can also set both the starting address and one or more breakpoints. Just include the breakpoint addresses on the same command line, separating them from the starting address and each other with spaces.

Keep this in mind: Before DEBUG executes a G command, it saves the values of all the registers, including the instruction pointer. If the program runs normally, and completes by executing INT 20H, DEBUG restores all the registers. This is great if your program runs all the way from beginning to end. You just type G and your program runs again. If, however, your program has just completed after being restarted from a breakpoint, the instruction pointer now points to the location where the breakpoint was set. Typing G starts it from the breakpoint again. To run the program from the beginning, type G =100.

Learning More About DEBUG

You've now used DEBUG to load and store program files, to assemble and disassemble a new machine language program, and to execute a program. Some other useful commands we don't have room to cover are D (Dump), which displays the contents of a block of memory as hexadecimal numbers and ASCII characters; E (Enter), to examine and change the contents of individual memory locations; and T (Trace), which executes an ML program one instruction at a time, displaying all registers and flags between instructions.

As you learn more about 8088 machine language, you'll find DEBUG a big help in testing your programs. Though you might use a separate assembler when your programs get larger, DEBUG remains useful for testing and modifying the assembled programs. If you want to know more, there is a complete description of each DEBUG command in Chapter 12 of the *DOS 2.00 Manual* and Chapter 8 of the *DOS 2.10 Manual*. Information on the DOS functions and interrupts can be found in Appendix D of the *DOS 2.00 Manual* and Chapter 5 of the *DOS 2.10 Technical Reference Manual*. To learn more about machine language programming on the IBM PC and PCjr, see COMPUTE!'s *Beginner's Guide to Machine Language on the IBM PC & PCjr.* ©

Save-With-Replace: Debugged At Last Part 2

P A Slaymaker

Last month, Part 1 proved that a long-suspected bug in the Commodore Save-with-Replace command really exists. Using a program that demonstrated the bug on a 1541 disk drive, the article showed how disks can be scrambled when files are scratched and rewritten with Save-with-Replace. The article also offered a brief explanation of the bug and how to avoid it. This month, Part 2 examines the Save-with-Replace bug in greater detail for technically advanced readers. The author is the president of Quantum Software, which produces the Peek a Byte disk utility for the Commodore 64.

What actually causes the Save-with-Replace bug? When and how does it occur and is there a fix for it? We have performed extensive testing to determine exactly how the bug happens. As explained last month, we've determined that the bug is avoidable if the drive number (drive 0) is specified in *all* disk commands. If you don't always specify drive 0, the bug occasionally bites. That's significant information in itself—but we wanted to know *why*.

DOS Thievery

First, we should note that although the SAVE@ command deletes a disk file and saves a replacement in a single operation, it works differently than if you issued separate SCRATCH and SAVE commands. SAVE@ calls entirely different DOS routines—the SCRATCH and SAVE are executed as part of a continuous procedure, and the SAVE@ command therefore requires that more drive buffers be available.

DOS V2.6 has five internal buffers, numbered 0 to 4. These buffers start at memory pages $300, $400, $500, $600, and $700, respectively. Normally an image of the disk's BAM (block availability map) is stored in the page at $700, an image of the directory sector in use is stored at $600, and the other three buffers are available for file use. As long as a buffer is active, it cannot be used for anything else. If DOS has assigned an internal channel to the BAM at $700, then trying to open a direct channel to buffer 4 (from BASIC: OPEN 2,8,2,"#4") will produce a 70,NO CHANNEL,00,00 error.

Similarly, DOS assigns channels and buffers to the directory sector and file sectors which are being read or written. Normally DOS assigns two read or two write channels and uses only three of the five buffers. The SAVE@ command, however, requires all five buffers—two read, two write, and the BAM. If DOS can't find a free buffer, it tries to steal an assigned but inactive buffer. This thievery causes the SAVE@ command to occasionally fail—for reasons which will be discussed shortly.

Why does omitting the drive number in disk commands cause DOS to steal a buffer? When a file is opened or loaded via the OPEN routine ($D7B4), DOS searches the internal directory to look for the specified filename (DOS routine names and addresses in this article conform to those listed in *Inside Commodore* DOS, Datamost, 1984). ONEDRV ($C312) determines whether a drive was specified. OPTSCH ($C3CA) assigns a default or specified drive for each file in the command, and also calls AUTOI ($C63D). AUTOI reads the BAM of the disk in the specified drive, and also tries to initialize drive 1 if no drive was specified. Usually buffer 3 ($600) is allocated for the phantom drive 1 BAM, and a B1 SEEK command is issued to the disk controller. This results in an internal DRIVE NOT READY error in the disk controller. The error is trapped by AUTOI but not reported outside the disk drive. This leaves buffer 3 allocated but inactive. FFST ($C49D) then reads the directory and tries to find the file.

The reason this inactive buffer assignment is important is that the SAVE@ command requires all five buffers, but only four are now available. Whenever DOS needs to allocate a buffer, it calls GETBUF ($D28E). If one is not free, GETBUF tries to steal an inactive one by calling STLBUF ($D339). If the drive number is always specified and no direct access buffers are allocated, STLBUF is never called. We verified this by modifying GETBUF after copying DOS onto an EPROM (Eraseable-Programmable Read Only Memory). If a channel can't be stolen, then a NO CHANNEL error occurs. But if STLBUF is called, the SAVE@ bug sometimes occurs.

Stealing The Wrong Buffer

STLBUF can be called several times during a SAVE@ command. The

COMPUTER MAIL ORDER

THE CMO ADVANTAGE

- THE BEST PRICES!
- Next day shipping on all in stock items.
- Free easy access order inquiry.
- Orders from outside Pennsylvania and Nevada save state sales tax.
- Free technical support with our factory trained technical staff.
- There is no limit and no deposit on C.O.D. orders.
- There's no extra charge for using your credit card. Your card is not charged until we ship.
- No waiting period for cashiers checks.
- We accept purchase orders from qualified corporations. Subject to approval.
- Educational discounts available to qualified institutions.
- FREE CATALOG MEMBERSHIP.

ORDER LINE

1-800-233-8950
In PA 1-800-242-4215

CUSTOMER SERVICE & TECH SUPPORT
1-717-327-1450
Dept. A211

MAILING ADDRESS

Computer Mail Order
Dept. A211
477 East Third Street
Williamsport, PA 17701

MEMBER DIRECT MARKETING ASSOCIATION

CREDIT CARDS

VISA MasterCard DINERS CLUB

SHIPPING

Add 3%, minimum $5.00 shipping and handling on all orders. Larger shipments may require additional charges.

All items subject to availability and price change.

Returned shipments may be subject to a restocking fee.

CANADIAN ORDERS

1-800-268-3974
Ontario/Quebec

1-800-268-4559
Other Provinces

1-416-828-0866
In Toronto

TELEX: 06-218960

2505 Dunwin Drive,
Mississauga, Ontario
Canada L5L1T1

All prices shown are for U.S.A. orders.
Call The Canadian Office for Canadian prices.

HOME COMPUTERS

ATARI

130XE (128K)	CALL
520ST (512K)	CALL
800XL 64K	CALL
850 Interface	$119.00
1010 Recorder	$49.99
1050 Disk Drive	$169.00
1025 Dot Matrix Printer	$199.99
1027 Letter Quality Printer	$269.99
1030 Direct Connect Modem	$59.99
8036 Atari Writer	$29.99
Star Raiders	$4.99
Missile Command	$4.99
Defender	$4.99
Galaxian	$4.99
Dig Dug	$4.99
Donkey Kong	$4.99
Joust	$4.99
Eastern Front	$4.99

While Supply Lasts!

BOARDS FOR ATARI

Axlon 32K	$39.99
Axlon 48K (400)	$69.99
Axlon 128K	$269.99
Bit 3 Full View 80	$169.00

APPLE

APPLE IIe	CALL
APPLE IIc	CALL
MacINTOSH	CALL
IIc LCD Display	CALL

Macintosh Software

Lotus Jazz	CALL
Microsoft Excel	CALL
Microsoft Business Pak	$375.00
Living Videotext ThinkTank 512	$159.00
Manhattan Ready, Set, Go	$79.99
Creighton Development Mac Spell	$69.99
Monogram Dollars & Sense	$99.99
Peachtree Back to Basics - GL	$109.00
PFS File & Report (New Version)	$129.00
Silicon Beach Airborn	$25.99

commodore

C128 Computer	$299.00
C1571 (Disk Drive for C128)	$NEW
C1902 (RGB 13" Monitor for C128)	$NEW
C1670 (Modem for C128)	$NEW
Commodore Plus 4	$199.00
CBM 64	$149.00
C1541 Disk Drive	$199.00
C1530 Datasette	$39.99
M-801 Dot Matrix Printer	$169.00
MCS 803 Dot Matrix	$179.00
C1702 Color Monitor	$189.00
C1660 Auto Modem	$59.99
DPS 1101 Daisy Printer	$339.00

Professional Software

Fleet System II w/Spell	$49.99
Trivia Fever	$29.99
Word Pro 4 Plus/5 Plus each	$239.00
Info Pro	$179.00
Administrator	$399.00

BRODERBUND

The Print Shop	$29.99

pfs:

File (64)	$59.99
Report (64)	$49.99

BATTERIES INCLUDED

PaperClip w/Spell Pack	$79.99
The Consultant DBMS	$69.99
Bus Card II	$139.00
80 Col Display	$129.00

PORTABLE COMPUTERS

HEWLETT PACKARD

41CV	$189.99
41CX	$249.99
HP 71B	$419.99
HP 11C	$62.99
HP 12C/15C/16C	$89.99
HP 75D	$949.00
HPIL Module	$98.99
HPIL Cassette or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

We stock the full line of HP calculator products

NEC

PC-8401	CALL
PC 8201 Portable Computer	$289.00
PC 8231 Disk Drive	$599.00
PC 8221A Thermal Printers	$149.00
PC 8281A Data Recorder	$99.99
PC-8201-06 8K RAM Chips	$105.00

SHARP

PC-1350	$159.99
PC-1261	$159.99
PC-1260	$109.99
PC 1500A	$165.99
PC-1250A	$88.99
CE 125 Printer/Cassette	$128.99
CE 150 Color Printer Cassette	$171.99
CE 161 16K RAM	$134.99

DISKETTES

maxell

3½" SS/DD	$39.99
3½" DS/DD	$54.99
5¼" MD-1 w/Hardcase	$17.99
5¼" MD-2 w/Hardcase	$23.99
5¼ MD-2-HD for AT	$44.99

Verbatim

5¼ SS/DD	$21.99
5¼ DS/DD	$29.99
Disk Analyzer	$24.99

Dennison

Elephant 5¼" SS/SD	$13.99
Elephant 5¼" SS/DD	$15.99
Elephant 5¼" DS/DD	$16.99
Elephant Premium DS/DD	$23.99

IBM

5¼" DS/DD Floppy Disks (Box of 10)	$25.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-n-File 10	$3.99
Flip-n-File 50	$17.99
Flip-n-File 50 w/lock	$24.99
Flip-n-File (400/800 ROM)	$11.99

AMARAY

50 Disk Tub 5¼"	$9.99
30 Disk Tub 3½"	$8.99

MODEMS

ANCHOR

Volksmodem	$59.99
Volksmodem 300/1200	$189.99
Mark XII (1200 Baud)	$259.00
Signalman Express	$299.00
Lightning 2400 Baud	$399.00

Hayes

Smartmodem 300	$139.00
Smartmodem 1200	$389.00
Smartmodem 1200B	$359.00
Smartmodem 2400	$629.00
Micromodem IIe	$149.00
Smart Com II	$89.99
Chronograph	$199.00
Transet 1000	$309.00

AST

Reach 1200 Baud Half Card	$399.00

MICROBITS

MPP-1000E AD/AA (Atari)	$79.99
MPP-1064 AD/AA (C 64)	$69.99

Novation

Smart Cat Plus	$319.00
J-Cat	$99.99
Novation 2400	CALL
Apple Cat II	$229.00
212 Apple Cat II	$379.00
Apple Cat 212 Upgrade	$229.00
Macmodem	$319.00

QUADRAM

Quadmodem II	
300/1200	$339.00
300/1200/2400	$499.00

TELELEARNING

C64 300 Baud (Closeout)	$39.99

GRAPHICS

Koala

IBM	$89.99
Apple/Franklin	$79.99

Polaroid

Palette	$1299.00

DRIVES

HARD

ALLOY

PC Slor	CALL

IBEL Associates

5 meg Removable/Internal	$1399.00
10 meg Fixed/Internal	$1249.00
15 meg 5 Removable/10 Fixed	$2149.00
25 meg 5 Removable/20 Fixed	$2499.00

I-MEGA

10 meg Bernoulli Box	$1999.00
20 meg Bernoulli Box	$2599.00
5 meg "MacNoulli"	$1499.00

TALLGRASS TECHNOLOGIES

COMMITTED TO MEMORY

12, 25, 35, 50, 80 meg (PC)	from $1499.00

FLOPPY

INDUS

Atari GT	$219.00
C-64 GT	$229.00

μ-SCI

A1.5 Apple	$179.00
A2 Apple	$179.00

MSD

SD1 C-64 Single	$219.00
SD2 C-64 Dual	$469.00

Tandon

320K 5¼" (PC)	$119.00

TEAC

320K 5¼"	$109.00

CALL TOLL-FREE

result is that the BAM and directory sectors can be reassigned to different buffers during a single SAVE@. We have found the BAM and directory sectors in every drive buffer after different SAVE@ commands. We have found copies of the current directory sector in two different buffers, one an old sector and one properly updated, but the wrong one had been written to the disk. Somehow, the pointers to the BAM and directory sectors are not properly accounted for. Which buffer is stolen by STLBUL depends on prior buffer usage and the values stored in LRUTBL,Y ($FA,Y), the least recently used table. It appears that STLBUF updates all pointers except LRUTBL,Y. This means that multiple calls to STLBUF may steal the wrong buffer—in this case the wrong buffer to steal is the BAM!

The BAM is stored in the drive in one of the buffers. STLBUF should not steal the drive 0 BAM, but should instead take back the unused buffer incorrectly assigned to drive 1. It never steals the drive 1 BAM, buffer 3 at $600, because STLBUF cannot take a buffer which encountered a drive error. Remember that an internal DRIVE NOT READY error did occur, because there is no drive 1!

To test this, we copied into EPROM an altered version of DOS with STLBUF modified to allow stealing a buffer with this error. This allowed the phantom drive 1 BAM buffer to be freed, and the SAVE@ bug did not strike during tests with this modified DOS.

If this buffer-stealing occurs, why does SAVE@ work most of the time? We must dig deeper into DOS to answer this question. When a file is opened and blocks (or sectors) are written to a disk, the BAM is *not* directly updated in the drive memory. Instead, a BAM image for each of two tracks is stored at BAM ($2A1–$2B0). Each time a new block is allocated by WUSED ($EF90), it is recorded in the BAM image. When a new track is tested for free sectors, DOS checks if it has a BAM image for it. If not, it calls SWAP ($F05B), which first updates the BAM with the BAM image from the next-to-last track, copies the new track's BAM map into the BAM image, and then zeros that track in the BAM. This all works perfectly—most of the time.

After the last file sector is written to the disk, the BAM still has not been written to the disk. In fact, the BAM in the drive is wrong because it has not yet been updated from the BAM images. When a file is closed, the disk directory is closed, CLSDIR ($DBA5), by reading in the file's directory sector, testing for a replace file type, and then rewriting it to the disk. MAPOUT ($EEF4) is called to read the BAM off the disk, if necessary, and to then update it from the BAM images by calling PUTBAM ($F0A5). The updated BAM is then written back to the disk.

During a SAVE@ command, DOS performs an additional step after reading the directory sector. The file type is designated as replace, so DELFIL ($C87D) is called to delete the original version of the file from the BAM. It reads in the BAM if necessary when freeing the first sector, FRETS ($EF5F), and then proceeds to trace through the file and delete sectors in the BAM images. The BAM is then written to the disk.

Bungled BAM

Normally this procedure works correctly. But havoc results if the BAM buffer is stolen while the file is being closed. This can happen during a SAVE@ command because DELFIL requires two additional buffers. The BAM can be stolen at different points during the procedure, depending on which buffers were previously used—which, in turn, depends on the number of sectors in the file and the tracks on which it is stored.

After the BAM is stolen, it is read back in when needed and updated from the BAM images. Only *two* tracks can be updated, however, since there are only two images. If more than two tracks have been accessed by SAVE@, the BAM may *not* be correctly updated. A track could be updated correctly, left unchanged, or fully allocated, depending on when the BAM was stolen.

If extra sectors are allocated, the BAM is incorrect, but no permanent harm is done. A validate command will cure the problem. If sectors are not allocated, then a new file will be saved on top of the old file's sectors. In the example program listed in Part 1, a fourth SAVE@ command would result in the file being written on top of the old file's first four sectors, and then the whole new file would be scratched—a tragic result, indeed.

Based on these findings, we recommend that you avoid the SAVE@ command when direct access channels to the drive are open or if you don't always specify the drive number in disk commands. You should also avoid SAVE@ when using programs or cartridges intended to speed up access on the 1541 disk drive. These programs often reserve internal drive buffers and may cause problems even if the drive number is specified. If you're using the DOS Wedge, we recommend issuing a >UI or >UJ command before each SAVE@ command to be sure all the buffer pointers are reset. Many word processors also allow you to send these commands to the drive. Otherwise, the drive should be turned off and then on before using SAVE@. (On the SX-64, press the drive reset button.)

During our studies we found several other minor bugs in DOS V2.6, including the subroutine which puts the value 2 at the drive memory location $197. This bug does no harm since it affects a normally unused section of drive memory. However, we have found it can affect DOS routines downloaded into the drive. There may be other bugs or quirks which we have not found, so the Commodore DOS controversy may never be fully closed.

In Part 1 of this article, there was a minor error in the example to illustrate the problems caused by not specifying a drive number (using the DOS Wedge program). The article stated that giving the Wedge command >$TEST results in a blinking disk error light if the file TEST does not exist on the disk. Actually, >$TEST does not cause the error light to blink unless it is used twice in succession. The first >$TEST correctly prints a blank directory of drive 0, but leaves the 1541 looking for the nonexistent drive 1 so that the second >$TEST results in the DRIVE NOT READY error described last month.

©

Atari REMover

Jeff Stefanski

This short BASIC utility automatically removes REM statements from programs. It runs on the Atari 400/800, XL, and XE series computers.

Many programmers use REMark statements to document how their programs work—a good programming practice. Once the program is finished and debugged, however, the REMs can be deleted to save memory and slightly increase execution speed (although it's a good idea to save a version with the REMs in case you have to make modifications later). Scanning through a program and deleting REM statements one by one has always been a tedious job. But it's easy with "Atari REMover."

This short routine automatically removes the REMs from BASIC programs, leaving everything else intact. Type in Atari REMover as listed below, then save the program by LISTing it to disk or cassette. *You must store the program with LIST, rather than SAVE.* (Example: LIST "C:" for cassette or LIST"D:*filename.ext*" for disk.) Since Atari REMover deletes itself from memory after running, be sure to save a copy before using it for the first time.

REMover is easy to use. First load the program from which you want to delete the REMs. Then append REMover to the end of the first program by ENTERing it from disk or cassette. (Example: ENTER "C:" for cassette or ENTER"D:*filename.ext*" for disk.) Type GOTO 32000 and press RETURN to activate REMover. The routine looks through your program and deletes each line that contains nothing but a REM statement. If a multistatement line ends with REM, the REM portion is cut off and the line is reentered.

It may take a while for REMover to delete all the REMs in a large program, so be patient. After the job is done, REMover deletes itself.

Note that REMover uses line numbers above 32000. If your program uses the same line numbers, renumber it before using this routine. If your program contains a GOTO or GOSUB to a REM line (poor programming practice in any case), change the line reference yourself after using REMover.

Atari REMover

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!

```
DD 32000 CLR :GRAPHICS 0:STM
         TAB=PEEK(136)+PEEK(
         137)*256:POKE 82,2:
         POKE 83,39:DIM L$(1
         14)
CL 32001 LINE=PEEK(STMTAB)+P
         EEK(STMTAB+1)*256
AK 32002 IF LINE=32000 THEN
         32015
FA 32003 PRINT CHR$(125):POS
         ITION 2,6:LIST LINE
HH 32004 LOCATE 3+LEN(STR$(L
         INE)),7,A:LOCATE 4+
         LEN(STR$(LINE)),7,B
         :LOCATE 5+LEN(STR$(
         LINE)),7,C
CA 32005 IF A=82 AND B=69 AN
         D C=77 THEN 32009
OP 32006 L=1:FOR X=7 TO 9:FO
         R Y=2 TO 39:LOCATE
         Y,X,M:L$(L)=CHR$(M)
         :L=L+1:NEXT Y:NEXT
         X
GI 32007 FOR X=1 TO 110:IF L
         $(X,X+3)=":REM" THE
         N 32012
ND 32008 NEXT X:STMTAB=STMTA
         B+PEEK(STMTAB+2):GO
         TO 32001
KH 32009 PRINT CHR$(125):POS
         ITION 2,6:PRINT LIN
         E
NC 32010 POSITION 0,0:POKE 8
         42,13:POSITION 2,7:
         PRINT "CONT":POSITI
         ON 2,4:STOP
LN 32011 POKE 842,12:GOTO 32
         001
IN 32012 PRINT CHR$(125):POS
         ITION 2,6:PRINT L$(
         1,X-1):PRINT "CONT"
FC 32013 POSITION 0,0:POKE 8
         42,13:POSITION 2,4:
         STOP
JF 32014 POKE 842,12:STMTAB=
         STMTAB+PEEK(STMTAB+
         2):GOTO 32001
BI 32015 PRINT CHR$(125):POS
         ITION 2,6:FOR X=320
         00 TO 32016:PRINT X
         :NEXT X:PRINT "PRIN
         T CHR$(125):POKE 84
         2,12:END"
BI 32016 POKE 842,13:POSITIO
         N 2,2:STOP
```

©

Plus/Term
For Commodore 1660 Modem

Mark Wood

By adding a few lines to COMPUTE!'s *popular "Plus/Term" program, you can use it with a Commodore 64 and the Commodore 1660 direct-connect modem, dialing and hanging up under program control.*

"Plus/Term," published in COMPUTE!, February 1985 (and in *Telecomputing on the Commodore 64*, COMPUTE! Books), is an excellent terminal program, offering an 80-column display mode (with "Screen-80," COMPUTE!'s GAZETTE, September 1984) and many other desirable features. However, since my Commodore 1660 is a direct-connect modem which doesn't allow manual dialing, I had no way to use Plus/Term. My solution was to add auto-dialing and hang-up routines to the program.

To include these new features in Plus/Term, you'll first need to type in the original program. Then type in the additional lines listed below. Once you're finished, resave the program (perhaps with a different name to distinguish it from the original Plus/Term).

Plus/Term now offers two additional options: You can dial a number from within the program (press D) or hang up the line whenever you want (H). After selecting Dial, type in the number you want, then choose between rotary and tone dialing, depending on which service you have on your phone system. You may add spaces or dashes between the numbers if you like, but they're not necessary. If you press RETURN without entering a number, or enter a string that contains no numbers, Plus/Term simply returns you to terminal mode. Rotary dialing is simulated by rapidly disconnecting and reconnecting the line the correct number of times for each number. Tone dialing signals are generated with the 64's SID chip.

Plus/Term Modifications

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!

```
10 DATA152,85,74,60,117,77,168
   ,44,152,85,168,44,161,94,16
   8,44,117,77,85,49  :rem 156
20 DATA152,85,85,49,161,94,85,
   49,117,77,150,54,152,85,150
   ,54,161,94,150,54  :rem 140
271 PRINTA$"H. HANG UP";
                        :rem 68
272 PRINTA$"D. DIAL"; :rem 152
281 IFM1$="H"THENPOKE56577,(PE
    EK(56577)OR32)      :rem 8
282 IFM1$="D"THEN2020 :rem 133
685 DIMDIG(15):FORWXE=0TO9:FOR
    COL=1TO4:READSID(WXE,COL):
    NEXT:NEXT           :rem 33
2020 ZX$="":AZ$="":INPUT"{CLR}
     {6 DOWN}{2 RIGHT}NUMBER T
     O DIAL";AZ$:IFLEN(AZ$)=0T
     HEN1760           :rem 120
2030 FORJ=1TOLEN(AZ$):G$=MID$(
     AZ$,J,1):IFG$=>"0"ANDG$<=
     "9"THENZX$=ZX$+G$:rem 190
2040 NEXT:IFLEN(ZX$)=0THEN1760
                        :rem 62
2050 PRINT"{DOWN}{2 SPACES}
     {RVS}R{OFF}OTARY OR {RVS}
     T{OFF}ONE?"       :rem 232
2060 GETTY$:IFTY$<>"R"ANDTY$<>
     "T"THEN2060       :rem 208
2070 FORWXE=1TOLEN(ZX$):DIG(WX
     E)=VAL(MID$(ZX$,WXE,1)):N
     EXT               :rem 154
2080 POKE56579,(PEEK(56579)OR3
     2):POKE56577,(PEEK(56577)
     AND223):FORJ=1TO600:NEXT
                       :rem 127
2090 PRINT"{3 DOWN}{5 RIGHT}DI
     ALING...":IFTY$="R"THEN21
     60                 :rem 71
2100 POKE54296,15:POKE54276,16
     :POKE54283,16:POKE54277,0
     :POKE54284,0       :rem 46
2110 POKE54278,240:POKE54285,2
     40:POKE54295,0:FORWXE=1TO
     LEN(ZX$)          :rem 236
2120 POKE54272,SID(DIG(WXE),1)
     :POKE54273,SID(DIG(WXE),2
     )                  :rem 13
2130 POKE54279,SID(DIG(WXE),3)
     :POKE54280,SID(DIG(WXE),4
     )                  :rem 23
2140 POKE54276,17:POKE54283,17
     :FORDEL=1TO75:NEXT
                       :rem 129
2150 POKE54276,16:POKE54283,16
     :FORDEL=1TO75:NEXT:NEXT:P
     OKE54296,0:GOTO1760:rem 9
2160 FORWXE=1TOLEN(ZX$):VA=DIG
     (WXE):IFVA=0THENVA=10
                        :rem 19
2170 FORCL=1TOVA:POKE56577,(PE
     EK(56577)OR32):FORDEL=1TO
     26:NEXT            :rem 78
2180 POKE56579,(PEEK(56579)OR3
     2):POKE56577,(PEEK(56577)
     AND223):NEXT      :rem 110
2190 FORDEL=1TO250:NEXT:NEXT:G
     OTO1760            :rem 99
```

©

Atari Animation With P/M Graphics
Part 3

Robert Powell

Animation with Atari player/missile graphics involves a number of programming techniques. Parts 1 and 2 in this series showed how to design a player/missile object, display it on the screen, control its color, and animate it horizontally. This month's article shows one method of vertical animation without resorting to machine language.

As we saw in Part 2, horizontal animation with player/missile graphics is quite simple: A single POKE into the horizontal position register moves the P/M strip to any place on the screen. Last month, Program 2 demonstrated how easy it is to move a player horizontally with a joystick.

Vertical animation, however, is not so simple. There is no such thing as a vertical position register which corresponds to the horizontal position register. Since P/M objects are strips of memory taller than the screen, a vertical register wouldn't make sense—you couldn't see the strip moving, anyway. Instead, to achieve vertical animation your program must move the P/M shape you've designed through the strip of player memory.

Program 1 below is a slightly modified version of last month's program which defined player 0 as a happy face. It shows how a shape can be moved through P/M memory with a FOR-NEXT loop in BASIC. Plug a joystick into port 1 to control the player's vertical movement. As you'll see, vertical animation in BASIC is disappointingly slow. BASIC just isn't fast enough to move the player shape through memory without a rippling "inchworm" effect.

There are two solutions to this problem. One is to write a machine language subroutine for vertical animation. Over the past five years, COMPUTE! has published several such routines which require no knowledge of machine language—you just drop the routine into your BASIC program and call it with a USR statement. The back issues are now out of print, but these and various other routines for vertical motion are discussed in several books (*COMPUTE!'s First Book of Atari Graphics; Mapping the Atari; COMPUTE!'s First Book of Atari Games;* and *COMPUTE!'s Second Book of Atari Graphics*).

Another solution which avoids machine language yet is comparable in speed takes advantage of BASIC's fast string-manipulation routines. We'll cover this method here.

A Few Strings Attached

The string-animation technique depends on making the computer think that a BASIC string is located in the P/M memory area, rather than in the usual memory area where the computer stores strings. Therefore, when you redefine the string, P/M memory changes—and the P/M object changes along with it. You can use this technique to rapidly change the shape of a player, move it vertically, or erase it off the screen.

Program 2 shows how to fool the computer into thinking a long string is located in the P/M memory area. A full explanation is beyond the scope of this article; however, even if you don't understand this technique, you can use it in your own programs by copying lines 10–100. This module adjusts itself for single- or double-line P/M resolution when you change the statement in line 20. Set MODE=1 for single-line resolution, or MODE=2 for double-line resolution.

When you run this program, several things become apparent. First, it eliminates the usual delay

Player/Missile Addresses Using String Animation

	Double-Line Resolution	Single-Line Resolution
Missiles 0–3	1–128	1–256
Player 0	128–256	256–512
Player 1	256–384	512–768
Player 2	384–512	768–1024
Player 3	512–640	1024–1280

caused by using a FOR-NEXT loop to clear out P/M memory with zeros. Instead, the three statements in line 100 clear out P/M memory instantly. This trick works by setting A$ to zeros after lines 10–90 fool the computer into thinking that A$ coincides with P/M memory.

Second, the program does not define the player shape by POKEing into P/M memory, as does Program 1 and last month's programs. Instead, the bytes which form the player shape in line 120 are read into a string (B$) in line 110. This is the key to the string-animation technique. Since the computer thinks that A$ overlays P/M memory, the statement in line 130 copies the player shape in B$ into the middle of the player 0 memory area. This places the shape at midscreen.

With a statement like A$(Y,Y+LEN(B$))=B$, you can instantly change the player's vertical position. For an example of vertical animation, replace lines 130 and 140 in Program 2 with the following lines:

```
130 FOR Y=256 TO 512
140 A$(Y,Y+LEN(B$))=B$
150 NEXT Y
160 GOTO 130
```

It's a convincing demonstration that fast vertical motion can be easily achieved in BASIC using strings.

Self-Erasing Players

If you look closely at the player shape bytes in line 120, you'll notice that a pair of zeros precedes and follows the series of numbers. Ordinarily, it doesn't make sense to see zeros in player shape data, because zeros show up blank on the screen. But these zeros have a special purpose. As the player shape moves through P/M memory, it would leave a trail of itself on the screen unless you erased it after every movement. Although it would be easy to erase the player shape by filling B$ with zeros (using a formula like the one in line 100), this extra step would slow down the animation by a fraction of second. By tacking a zero onto each end of the player data, the shape erases itself as it moves.

In this case, two zeros surround the player data. This allows even faster vertical motion by moving the player shape *two* steps at a time. To see this in action, add the above changes to Program 2 with this alteration to line 130:

```
130 FOR Y=256 TO 512 STEP
    2
```

Now change STEP 2 to STEP 15. As you can see, you can have as many shapes displayed in the vertical band as will fit.

Another important advantage of string-animation is that you can store several different shapes in different strings (such as B$, C$, D$, and so on). You can instantly flip between the shapes simply by reassigning A$, as in A$(Y,Y+LEN(D$))=D$.

What About Diagonals?

Once you learn how to move P/M objects horizontally and vertically, it's easy to animate them diagonally as well. Just combine a horizontal step with each vertical step, interweaving them to achieve a diagonal path.

For an example, start with Program 2 and add these changes:

```
130 FOR Y=256 TO 511
140 A$(Y,Y+LEN(B$))=B$:PO
    KE 53248,Y-256
150 NEXT Y
160 GOTO 130
```

If you experiment with these programs, you should be able to take it from here. All these examples use player 0, but the other players and missiles can be used in a similar manner. Just calculate the vertical screen position by figuring where A$ overlaps the appropriate player/missile area, then position the player shape data at that point in A$. (Refer to the accompanying table for a guide.)

Trying drawing a background screen with PLOT and DRAWTO, then move your players above or beneath it. Also, although P/M graphics are commonly used for games, try using these techniques to add interest and variety to your text programs as well. You can turn players or missiles into thin vertical lines to delineate data columns, or change them into cursors that change color to signal for input. This three-part series merely covers the basics—there's a lot more to Atari P/M graphics, such as priority registers and collision registers. The possibilities are endless.

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!

Program 1: Vertical Movement With FOR-NEXT

```
NF 10 POKE 106,PEEK(106)-8
NF 20 POKE 54279,PEEK(106)
HD 30 GRAPHICS 0:SETCOLOR 2,
      0,0
CN 40 PMBASE=PEEK(106)*256
ML 50 POKE 559,62
PM 60 POKE 53277,3
DP 70 POKE 704,68:POKE 705,1
      98:POKE 706,168:POKE 7
      07,148
PA 80 POKE 53248,160:POKE 53
      249,170:POKE 53250,180
      :POKE 53251,190
MB 90 FOR X=PMBASE+1024 TO P
      MBASE+2048:POKE X,0:NE
      XT X
LB 95 VERTICAL=PMBASE+1152
KD 100 RESTORE :FOR X=1 TO 1
       3
DP 110 READ A
AM 120 POKE VERTICAL+X,A
CL 130 NEXT X
LP 140 DATA 0,24,60,126,90,2
       19,255,219,195,102,60
       ,24,0
CF 150 S=STICK(0)
JK 160 IF S=13 THEN VERTICAL
       =VERTICAL+1:GOTO 100
JD 170 IF S=14 THEN VERTICAL
       =VERTICAL-1:GOTO 100
GD 180 GOTO 100
```

Program 2: Vertical Movement With Strings

```
BL 10 DIM A$(1),B$(15)
DA 20 MODE=2:REM MODE=2 FOR
      DOUBLE-RES, MODE=1 FOR
       SINGLE-RES
FH 30 PMPAGE=PEEK(106)-4*MOD
      E:POKE 106,PMPAGE:POKE
       54279,PMPAGE
CE 40 GRAPHICS 0:SETCOLOR 2,
      0,0:PMBASE=256*PMPAGE
EH 50 POKE 559,30+16*MODE:PO
      KE 53277,3:POKE 53248,
      160:POKE 704,68
HH 60 VTAB=PEEK(134)+256*PEE
      K(135):ATAB=PEEK(140)+
      256*PEEK(141)
CD 70 OFFSET=PMBASE+384*MODE
      -ATAB:REM FIND DISTANC
      E FROM ATAB START TO P
      LAYER ZERO START
FB 80 HI=INT(OFFSET/256):LO=
      OFFSET-256*HI:L=640*MO
      DE:HL=INT(L/256):LL=L-
      256*HL
AM 90 POKE VTAB+2,LO:POKE VT
      AB+3,HI:POKE VTAB+6,LL
      :POKE VTAB+7,HL
JD 100 A$(1)=CHR$(0):A$(640*
       MODE)=CHR$(0):A$(2)=A
       $
BB 110 FOR I=1 TO 15:READ A:
       B$(I,I)=CHR$(A):NEXT
       I
HF 120 DATA 0,0,24,60,126,90
       ,219,255,219,195,102,
       60,24,0,0
KM 130 A$(190*MODE,190*MODE+
       LEN(B$))=B$
GD 140 GOTO 140
```

©

Amiga's *Amazing* Graphics

Charles Brannon, Program Editor

Commodore's Amiga presents programmers with more graphics features than ever before—both an exciting prospect and a bewildering abundance. This overview covers the fundamentals of the Amiga's graphics capabilities and shows how they differ from those on previous personal computers.

Graphics make the Amiga special. Although the Amiga's other features—such as its stereo sound, high-speed 68000 microprocessor, built-in 880K disk drive, and multitasking operating system—are certainly noteworthy, it's the graphics that first catch your eye. The 4,096 color variations allow nearly seamless transition between colors; the 640 × 400 high-resolution bitmap mode is close to broadcast TV quality; and the custom chips permit fast, complex animation. The Amiga is a machine for the artist in all of us.

Making the most of these features requires programmers to master some new techniques, however. There are some important differences between the way the Amiga handles graphics and the methods used on previous personal computers. Of course, there are many similarities, too.

Mixing A Rainbow

Understanding any computer's graphics is easier if you know some background about video displays. In any monitor or TV, video images are electronically painted by electron guns on the inside of the CRT (*cathode ray tube*, commonly known as the picture tube). From our point of view when looking at the screen, three electron beams sweep left to right, top to bottom, across the inside of the CRT. The CRT is coated with special phosphors that glow either red, green, or blue when hit by the stream of particles from the electron guns. Each phosphor dot can glow bright, dim, dark, or anywhere in between.

Once painted on the screen, the video image quickly fades away, so the electron beams repeat the cycle to draw a new frame 60 times per second. This *refresh rate* is more than fast enough to fool our eyes into seeing motion when the video images are changing each frame, as they are with TV shows and animated computer graphics.

Unlike most computers, the Amiga does not limit you to a fixed set of colors. Instead, you mix three primary colors—red, green, and blue—to create your own custom colors. Each primary color has 16 luminance, or brightness, levels, from 0 (no color) to 15 (very bright). This makes up to 4,096 combinations possible (16 * 16 * 16 = 4,096).

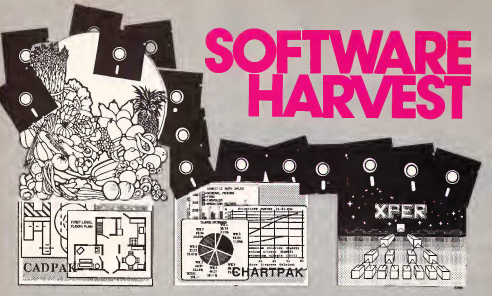
To display all these colors, the Amiga requires a special monitor called *analog RGB* (red-green-blue). The Amiga also works with another type of RGB monitor, called *digital RGB*, but these devices can display only 16 colors. (Digital RGB monitors are the type used with IBM computers and the Commodore 128.) If you plug the Amiga into a regular TV via its built-in RF modulator, 3,616 colors can be displayed.

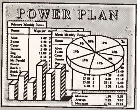
With a maximum of 4,096 color combinations available using analog RGB, almost any hue can be closely approximated. Most colors can be separated into red, green, and blue components. Because video images appear as transmitted, not reflected light, the red-blue-yellow primary color mixing you may have learned does not always apply. For example, red, blue, and green combine to form white, not dark brown. To get brown you'd need to combine red and green to get a greenish red color which appears to be brown. Turning up the brightness of green and red gives yellow. Combine dark red and dark blue to get violet. Bright red and dark blue yields a pastel shade of purple.

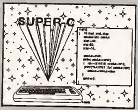
Color Indirection

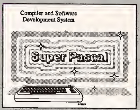
In most of the Amiga's graphics modes, you cannot display all 4,096 colors simultaneously. Instead, you're limited to a palette of 16 or 32 colors, depending on the mode. However, you can choose which of the 4,096 colors will be available in the palette.

SOFTWARE HARVEST









Cadpak Enhanced Version - Graphics design and drawing package. Create detailed designs with dimensioning, scaling, text, rotating, more. Use with or without lightpen. Hardcopy. $39.95

PowerPlan - *"one of the best programs ever written for the C-64"* Commodore Magazine, Nov 1985. Complete spreadsheet with builtin graphics. Cell formatting, protection, windowing, HELP. $39.95

Chartpak - Make professional quality pie, bar and line charts/graphs directly from your data. Instantly draw or redraw charts to your specifications. Statistics capabilities. Hardcopy. $39.95

Super C - Complete **C** language development system. Enhanced full-screen editor, compiler, linker, library and extensive handbook. Used by thousands in schools and industry. $79.95

Xper - First expert system for C-128 and C-64. Ordinary data bases play back facts, but XPER's *knowledge base* help you identify, diagnosis and select - in short helps makes decisions. $59.95

Super Pascal - Complete Pascal language development system. Full-screen editor, compiler, integrated assembler, graphics toolkit, fast DOS (3X faster than 1541) and handbook. $59.95

Call now, for the name of your nearest dealer. Or order directly from ABACUS with your Mastercard, VISA or AMEX card. Add $4.00 for postage and handling. Foreign orders add $10.00 per product. Other software and books are also available. Call or write for free catalog. Dealer inquiries welcome - over 1200 dealers nationwide.

Abacus Software

P.O. Box 7211 Grand Rapids, MI 49510

Tel. 616/241-5510 Telex 709-101

The colors in the Amiga palette are determined by 32 memory registers. Each register is 12 bits wide (1½ bytes), the number of bits needed to hold a number from 0 to 4,095.

When the Amiga draws a video image, the dots that make it up derive their colors from the color registers. In the 320 × 200 or 320 × 400 modes, each dot on the screen can be colored from any of the 32 color registers. Therefore, 32 simultaneous colors are possible in these modes.

Some computers, such as the Commodore 64, store color information for the screen in a section of memory known as *color memory.* Color memory is a grid of memory cells. Each cell defines the color of an 8 × 8 pixel zone. The number in color memory is a number representing 1 of 16 fixed colors. Other computers, such as the Atari machines, store color information by another method known as *color indirection.* The Amiga uses the latter technique. The number representing the dot on the screen does not encode the color directly, but instead selects a particular color register. The dot gets its color indirectly through the color register. If you change the color register, everything drawn with that register instantly changes.

Color indirection is extremely powerful. Unique glowing effects are possible by cycling between all the colors at high speed. You don't have to redraw the entire screen, which takes a lot of time. You can merely change a color register to instantly modify the appearance of everything drawn in that color. A single memory change affects an entire screen, which makes possible some high-speed effects even in a relatively slow language like BASIC.

For example, if you draw a series of concentric circles, each circle deriving its color from a different color register, you could create a 3-D tunnel illusion by changing the color registers in sequence. You could fill all color registers with the same color, then change one color register at a time to create the illusion of growing circles. Objects can be made invisible by changing their color to the same color as the screen background, then made to appear instantly by giving them a contrasting color.

A Nybble Of Color

Color information is stored in the color registers by flipping certain bits on and off. Each 12-bit register assigns 4 bits for each primary color. (A group of 4 bits is called a *nybble*—half of a byte.) In the 320 × 200 and 320 × 400 modes, here is the format of the color registers:

11 10 9 8	7 6 5 4	3 2 1 0	*bit number*
blue	green	red	*primary color*

A handy formula for setting a color register in this mode is:

BLUE*256+GREEN*16+RED

where the luminance values of BLUE, GREEN, and RED range from 0–15.

The color registers for the hires 640 × 400 mode with an analog RGB monitor are a little trickier:

11 10 9	8 7 6	5 4	3	2 1 0	*bit number*
blue	green	rm rl	rh	b g r	*color*

As you can see, the color bits have been scattered all over the 12-bit range. Bits 9–11 define 3 bits of data for blue (range of 0–7); bits 6–8 define 3 bits of data for green (range 0–7). Bits 4 and 5 are the low and medium bits of data defining red, and bit 3 is the high bit of red data. (You would think the red bits would be arranged high-medium-low instead of high-low-medium, but the Amiga engineers must have had some reason for this strange order.) Bits 0–2 are the enable bits for the red, green, and blue electron guns.

The formula for setting a color register in this mode is also more complicated:

BLUE*512+GREEN*64+INT(RED/2)*16+(RED AND 4)*2+BEN*4+GEN*2+REN

This formula assumes RED, GREEN, and BLUE range from 0–7; REN, GEN, and BEN (the RGB enable bits) are either 0 or 1; that INT takes the integer result of its argument (as in BASIC); and that AND performs a bitwise AND.

A Binary Tower

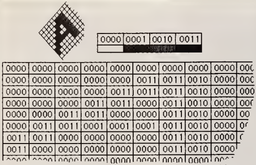
Each screen dot, or *pixel*, derives its color from one of these color registers. How are these dots laid out in memory? For a 32-color mode, each pixel is represented by a five-bit binary quantity ($2^5=32$). However, a five-bit quantity does not pack into a byte very well. Therefore, the Amiga maps its screen memory in a different way from most computers.

Traditionally, computers have laid out their screen memory serially, left to right, top to bottom. For instance, the Commodore 64's multicolor graphics mode fits four pixels into a byte, with each bit pair representing one pixel (00=color 0, 01=color 1, 10=color 2, 10=color 3). That's why the Commodore 64's 160 × 200 multicolor mode requires 8K of screen memory. With this memory scheme, to get more colors you would have to group more pixels together. But with five bits needed to store a single pixel on the Amiga, three bits would be wasted in every byte. If the Amiga used a serial scheme to store its display, it would take 64K to hold a 320 × 200 screen with 32 colors.

This problem was solved by grouping the bits a different way. Instead of using horizontally adjacent bits within the same byte to select a color register, the Amiga overlays bytes and reads the bits vertically. For example, all bits in bit position 7 from each of five overlaid bytes form a five-bit quantity. It's as if each pixel were a five-bit tower rising above the screen map. If you cross-section the vertical bytes making up the screen, you get five layers of bits called *bit planes.*

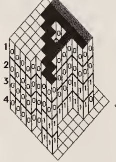
Each bit plane permits one bit of color definition. The simplest screen has only one bit plane, with one bit per pixel. This arrangement permits only on/off possibilities for each pixel. To get a broader range, you need to add another bit plane. That way, the bit on the primary bit plane and the bit in the corresponding position in the second bit plane permit two bits, or four possibilities of color definition. The accompanying figure shows how the Amiga uses bit planes for color selection, and the table gives a summary of the Amiga's screen modes.

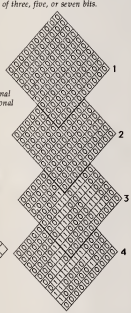
Incidentally, the Amiga has no true text modes like those found on earlier computers. Text is drawn as graphics objects, usually in 640 × 200 graphics, the default mode used by the Amiga's operating system, Intuition.



This shows how screen memory is mapped serially. Every pixel is represented by a four-bit binary number. We can pack two pixels per eight-bit byte. This scheme is not efficient for packing pixel sizes of three, five, or seven bits.

Below, the pixels are represented as "binary towers." The planes on the right show the three-dimensional figure separated into two-dimensional bit-planes.





A more flexible and efficient way of representing multiple-bit objects (pixels) is to layer the bits "vertically." Each bit-plane is equivalent to one high-resolution screen. A pixel is represented by a single bit position, hence eight pixels per byte. To permit a pixel to represent more than just on or off, additional bit fields are layered. All bits in a corresponding bit position together define a four-bit value.

Amiga Screen Modes

Mode	Pixels	Bit Planes	Memory	Onscreen Colors
Normal res, noninterlaced	320 × 200	1 to 5	8K–40K	2 to 32
Normal res, interlaced	320 × 400	1 to 5	16K–80K	2 to 32
Hi res, noninterlaced	640 × 200	1 to 4	16K–64K	2 to 16
Hi res, interlaced	640 × 400	1 to 4	32K–128K	2 to 16
Hold and modify	320 × 200	5 or 6	48K	256 to 4,096
Hold and modify	320 × 400	5 or 6	96K	256 to 4,096

Bit Planes Save Memory

Although the bit planes are stacked together, don't think of them as multiple, transparent screens. The pile of bit planes creates only one screen. Bit planes are merely a way to make more efficient use of memory. If you don't need 32 colors, use fewer bit planes. Since each bit plane uses 8K in 320 × 200 resolution, memory usage can range from 8K (one bit plane, 2 colors) to 40K (five bit planes, 32 colors). Memory usage for the 320 × 400 mode can be up to 80K; for 640 × 200, 64K; and 640 × 400 uses as much as 128K. Again, these are only if all allowed bit planes are used. The mininum memory requirement for 640 × 200 is 16K, for a two-color mode.

The Amiga permits up to six bit planes, though only five are used at a time in 320 × 200 or 320 × 400 resolution. The hi-res 640 × 200 and 640 × 400 modes use only four bit planes, for a total of 16 colors. Although six bit planes are available, the video circuitry can't fetch more than five bits per pixel in 320 × 200 mode, or it would lag behind the speeding video beam. The video beam that refreshes the picture can't wait, so the video circuitry must keep up with the beam.

That's why 640 × 200 and 640 × 400 modes are limited to four bit planes, or 16 onscreen colors. The video circuitry just can't fetch memory fast enough to change twice as many pixels per line. In fact, adjacent pixels cannot always have different colors in the 640 modes. The adjacent values may point to different color registers, but it's not possible to fully change the hardware color output between even and odd pixels in the 640 mode. A dark green pixel to the right of a bright white one may appear gray ("dark white"). Only the luminance information can be effectively changed before the beam has left that pixel position.

Interlaced Modes

Most personal computers up to now have been limited to a vertical resolution of 200 scan lines (a *scan line* is the thin horizontal line painted on the CRT by the electron guns as they scan from left to right). However, the Amiga can make use

of *interlacing* to double the number of scan lines. A TV or monitor displays more than 400 visible scan lines, but normally uses only every other scan line of a screen, filling in the odd lines with data from the even lines. In interlace mode, alternate screen refreshes are shifted up or down by one scan line, permitting full vertical resolution.

Interlacing on the Amiga works like this: In 1/60 second, the first 320 × 200 or 640 × 200 screen is scanned. Before the phosphors fade, a second 320 × 200 or 640 × 200 screen is scanned, shifted down one line to interweave it with the previous screen. The first screen displays even lines, and the second displays the odd lines. The result is doubled resolution—400 scan lines instead of 200.

Because the total picture takes twice as long to display, the phosphors in the even lines begin to fade as the odd lines are drawn. Therefore, some flickering and jittering in the 640 × 400 mode is visible. The only way to avoid this would be to redesign the monitor to refresh its screen at a faster rate than 1/60 second (a technique used by the Macintosh and Atari ST monochrome monitors).

The Amiga's video chip is smart enough to handle interlacing with ease. The bit planes are laid out in memory as if there were just one continuous 320 × 400 or 640 × 400 screen. You tell the video chip how far to skip ahead in memory to display the next line. By choosing an offset twice the normal line width, you can make the chip skip the odd lines of data for the first scan, then display the next screen from the odd lines, skipping the even lines. This greatly simplifies screen layout. The operating system actually takes care of these details, so you needn't even know how 320 × 400 or 640 × 400 are supported.

Dual Playfield Mode

In the normal-res modes, you can set up and overlay two independent graphics screens. A portion of one screen can be transparent to show the underlying screen. You can specify which screen has priority over the other. Each screen can be manipulated independently, even resized and moved over or under the other.

The overlay screens can use up to three bit planes each, since there are six available. However, you can use fewer bit planes if you want to save memory at the expense of color selection. Three bit planes permit only eight colors per screen, but each screen has its own color palette. And the palettes can contain any of the 4,096 hues, of course.

Dual playfields raise some exciting possibilities. In a game, one screen could show your cockpit window or starship control panel, with the windshield or viewport simply a transparent hole. The secondary screen could show your view of the sky or of the depths of space, visible through the transparent part of the primary screen.

Intuition uses this feature to let you slide down the top screen to see another behind it. For business applications, you could have two spreadsheets or documents running on the two screens simultaneously. Each screen can have its own windows, too.

Fine Scrolling

Scrolling is a technique that lets an *actual screen* pan across a much larger *virtual screen*. The actual screen is what you see on the monitor; the virtual screen includes the portions which won't fit on the monitor but can be scrolled into view. Scrolling lets you work with a very large document, spreadsheet, or page of graphics, and also makes for exciting computer games (such as *Defender* or *Eastern Front*).

Some computers are limited to *coarse scrolling*—the actual screen can be scrolled over the virtual screen only in character-sized increments. *Fine scrolling* is a more difficult technique that scrolls the actual screen pixel by pixel.

Fine scrolling is easy on the Amiga. The start of the screen map is found in two memory registers which are bit plane pointers. To scroll the screen up, just change the registers to point one line higher in memory. To scroll down, you subtract the line width from the bit plane pointers, displaying from the previously off-screen line of data.

For horizontal scrolling, a single register lets you shift the screen by up to 16 positions. You must fetch an extra data word per line to provide the pixels that should appear as the screen is scrolled. After you've scrolled 16 times, the program must perform a coarse scroll by repointing the bit plane pointer to the next word of memory. The whole display appears to move, but you're really just displaying a different section of memory.

4,096 Colors At Once

A special video mode lets you display more than just 32 colors at a time. *Hold and modify* mode can display 4,096 colors simultaneously in the normal-res modes.

It's a difficult mode to use, though. Each pixel is defined as a modification of the color of the previous pixel. You can hold this value, and modify a portion of it (hence *hold and modify*). Instead of bit plane data defining a color register, the bits from bit planes 5 and 6 determine which portion of the previous color output should be modified, and the bits from bit planes 0–4 are substituted in the selected portion of the color output.

You can define an entire screen of color, even the background color, just by modifying a single color register in this mode. You could start with bright white, then set the blue bits to zero to select yellow. From yellow you can decrease the red level to get green. You could then turn off the green bits to get black, which can in turn be modified to get bright blue. You can modify only the R, G, or B portion, or start over with data from a new color register.

Since color in the hold and modify mode is dependent on previous values, changing one pixel could change the colors of all following ones. It's a difficult mode to use for dynamic displays, so it is best suited for static pictures that need 4,096 colors.

There's much more to the Amiga's graphics than we can cover in this article. We haven't even begun to discuss blitter animation, sprites, the copper (video coprocessor), and options like external video mixing, video digitizing, and frame-grabbing. It will probably be a while before programmers learn to take advantage of all these features. In the meantime, we'll have a lot to look forward to. ©

A Better Way To POKE On The Commodore 64

Matthew MacKenzie

This "program-writing program" for the Commodore 64 can speed up any BASIC routine that uses POKE to fill large areas of memory. Using clever programming techniques, it writes a new routine that employs fast PRINT statements in place of POKEs.

BASIC programs often require that you fill a certain memory area with data. The data may be a machine language routine, sprite shape definitions, a high-resolution graphics screen, or whatever. In most cases, the job is done with POKE statements: The program READs values from DATA statements and POKEs each value into the computer's memory. Unfortunately, POKE is one of BASIC's slower statements. In fact, it's so slow that some programs display countdown timers during memory-filling operations to tell you how much longer you'll have to wait.

PRINT, on the other hand, is very fast. Though it's intended for a different purpose than POKE, PRINT also puts new values into certain memory locations. After all, the screen is just another memory area in the computer: It consists of 1000 locations numbered from 1024 to 2023, with 1024 at the upper-left corner. When you PRINT the letter *A* in the upper-left corner of the screen, you're storing a new value in memory location 1024. Because of this similarity, it's possible to store values in memory with PRINT instead of POKE.

However, PRINT's memory-changing ability has certain limitations. Usually, you can print only in the 1000-character screen memory area. And after you've changed character 999, the screen begins to scroll. The top line of the screen disappears, and everything below that line moves up. Finally, POKE and PRINT use different codes to represent characters, requiring conversion from Commodore ASCII (for PRINT) to screen code values (for POKE).

POKEing With PRINT

"Print Poker" solves all those problems. If you already have a routine that uses POKEs to fill memory, Print Poker can write a new, faster routine that does the same job with PRINTs. You don't have to understand how the special PRINT technique works to use Print Poker; it automatically creates new BASIC program lines containing everything you need. Type in the program listed below, and then save it. Because this program does some unusual things, be sure to read the following instructions *before* you try to run it.

As an example, say you've written a routine that puts eight sets of sprite shape data in memory locations 12288–12798. Your routine works fine, but POKEing the data into those 511 locations causes a noticeable delay. The Print Poker program can write a new BASIC routine that uses PRINT to do the same job more quickly. Before you run it, however, *you must run your own routine* to put the sprite data in memory. (Print Poker works only when the needed data is already in the proper memory area.)

Once that's done, enter NEW. Now you can load and run Print Poker. The program first asks you for beginning and ending addresses. In this case, you want the special PRINT statements to fill locations 12288–12798, so you enter 12288 for the beginning address and 12798 for the ending. This tells the program which memory area to look at when creating the special PRINT statements.

Next, you're asked to enter a starting line number. This is the

first line number of the routine Print Poker is about to write for you. Use whatever line number is appropriate for your routine. Since Print Poker itself uses line numbers from 60000 to 60380, use numbers considerably below 60000 to prevent a conflict. The program also asks you for a line increment. Since you won't have any reason to edit the new routine after it's made, line increments of two or five are fine.

Finally, you're asked whether Print Poker should delete itself when it's finished. If you're creating only one new routine, press Y to answer yes. Press N for no if you're creating two or more sets of special PRINT statements in a single session. (Use RUN 60000 to run Print Poker a second time. You should always delete Print Poker the last time it's used.)

One Program Writes Another

After you answer all the prompts, Print Poker goes to work, using the *dynamic keyboard* technique to write each line of your new routine. The program itself is writing another program. First, it puts a line number and the needed characters on the screen. Then it stores the line in BASIC memory just as if you'd moved the cursor to that line and pressed RETURN. When large memory areas are involved, this may take a couple of minutes. After the program stops and the blinking cursor reappears, your new routine is complete, ready to be saved and incorporated into a program.

The special PRINT statements look quite strange, of course. Because POKE can take any value from 0 to 255, the equivalent PRINT statement is usually a collection of graphics characters, including some nonprinting character values like CHR$(2). Such lines are difficult if not impossible to edit. Thus, it's best to use Print Poker only when your data is in final form (after you've finished making changes in the sprite shapes and so on). If you must make a change, you'll find it much easier to change and rerun the POKE version of your routine, and then run Print Poker again.

It took some creative programming to overcome PRINT's limitations. The value stored in location 648 tells the 64 where in memory PRINT should put its data. By carefully manipulating the value in 648, you can divert PRINT's output to any memory location in the computer and defeat screen scrolling as well. When Print Poker has finished its work, it sets everything back to normal with POKE 648, 4. Note that this technique does not work correctly in the highest 256 bytes of memory used by BASIC (locations 40740–40959). Use the conventional POKE method to put data in those locations. In addition, if you intend to put data at the top of BASIC user space, with Print Poker or without it, remember to move down the top-of-BASIC and top-of-string storage pointers to protect your data.

Print Poker

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
60000 POKE53281,14:PRINT"{CLR}
      {DOWN}{BLK}{9 SPACES}P R
       I N T{3 SPACES}P O K E
      {SPACE}R"             :rem 6
60010 INPUT"{5 DOWN}STARTING A
      DORESS";S            :rem 32
60020 INPUT"{2 DOWN}ENDING ADD
      RESS";E              :rem 41
60030 INPUT"{2 DOWN}STARTING L
      INE NUMBER";L       :rem 211
60040 INPUT"{2 DOWN}LINE INCRE
      MENT";I              :rem 65
60050 INPUT"{2 DOWN}SELF- DEST
      RUCT (Y/N)";D$      :rem 128
60060 V1=INT(S/256):V2=S-256*V
      1:IFLEFT$(D$,1)="Y"THEND
      =1                  :rem 209
60070 PRINT"{CLR}"L"
      {11 SPACES}P1=PE(648):PO
      648,"V1":A$=CH(34)+CH(34
      )+CH(20)"            :rem 31
60080 L=L+I:PRINT:PRINTL"
      {11 SPACES}?"CHR$(34)"
      {HOME}"CHR$(34)";";:IFV2
      =0THEN60100          :rem 11
60090 PRINT":FORX=1TO"V2":?"CH
      R$(34)"{RIGHT}"CHR$(34)"
      ;:NE":PRINT:RL=60110:GOT
      O60370              :rem 133
60100 PRINT:PRINT:RL=60110:GOT
      O60370              :rem 203
60110 PRINT"{CLR}"L"PRINT"CHR$
      (34);:PS=1024+POS(X):FOR
      X=1TO12:PRINT"{6 SPACES}
      ";:NEXT             :rem 192
60120 PRINT:PRINT:FORX=0TO3:A$
      (X)=MID$(CHR$(34)+"{A}$"
      +CHR$(34),X+1,1):NEXT
                          :rem 196
60130 S1=PEEK(S):IFS=ETHEN6024
      0                    :rem 77
60140 S1=PEEK(S):IFS1AND128AND
      R=0THENPOKEPS,146:PS=PS+
      1:R=1:GOTO60160      :rem 29
60150 IF(S1AND127)=S1ANDR=1THE
      NPOKEPS,210:PS=PS+1:R=0
                          :rem 243
60160 IFS1=34ORS1=162THENFORX=
      0TO3:POKEPS+X,ASC(A$(X))
      :NEXT:PS=PS+3:GOTO60180
                           :rem 87
60170 POKEPS,(S1AND127):rem 62
60180 IF(S+1)/256=INT((S+1)/25
      6)THENS2=INT(S/256)+1:GO
      TO60220             :rem 180
60190 IFPS>1090THEN60210
                           :rem 98
60200 PS=PS+1:S=S+1:GOTO60130
                          :rem 189
60210 POKEPS+1,34:POKEPS+2,59:
      RL=60110:S=S+1:GOTO60370
                          :rem 182
60220 POKEPS+1,34:POKEPS+2,59:
      L=L+I:PRINT:PRINT:PRINTL
      "PO648,"S2":?:?:?";
                          :rem 130
60230 PRINTCHR$(34)"{HOME}"CHR
      $(34)";":RL=60110:PRINT:
      S=S+1:GOTO 60370:rem 144
60240 POKEPS+1,34:POKEPS+2,59:
      PRINTL+I"{11 SPACES}";
                          :rem 198
60250 PRINT"PO648,P1:?"CHR$(34
      )"{HOME}"CHR$(34):RL=-1:
      IFDTHENRL=60280       :rem 8
60260 GOTO60370            :rem 55
60270 RL=60300:PRINT"{CLR}6000
      0":PRINT"60010":PRINT"60
      020":PRINT"60030"
                          :rem 112
60280 PRINT"60040":PRINT"60050
      ":PRINT"60060":PRINT"600
      70{DOWN}":GOTO60380
                          :rem 101
60290 RL=60320:PRINT"{CLR}6008
      0":PRINT"60090":PRINT"60
      100":PRINT"60110"
                          :rem 130
60300 PRINT"60120":PRINT"60130
      ":PRINT"60140":PRINT"601
      60{DOWN}":GOTO60380
                           :rem 91
60310 RL=60340:PRINT"{CLR}6017
      0":PRINT"60180":PRINT"60
      135":PRINT"60200"
                          :rem 133
60320 PRINT"60210":PRINT"60220
      ":PRINT"60230":PRINT"602
      40{DOWN}":GOTO60380
                           :rem 92
60330 RL=60360:PRINT"{CLR}6025
      0":PRINT"60260":PRINT"60
      270":PRINT"60280"
                          :rem 143
60340 PRINT"60290":PRINT"60300
      ":PRINT"60310":PRINT"603
      20{DOWN}":GOTO60380
                           :rem 99
60350 RL=-1:PRINT"{CLR}60330":
      PRINT"60340":PRINT"60350
      ":PRINT"60360"  :rem 236
60360 PRINT"60370":PRINT"60380
      ":PRINT"60390":PRINT"601
      05{DOWN}"             :rem 1
60370 PRINT"S="S"{LEFT}:E="E"
      {LEFT}:L="L+I"{LEFT}:I="
      I"{LEFT}:D="D"{LEFT}:R="
      R;                  :rem 251
60380 PRINT"{LEFT}:GOTO"RL"
      {DOWN}":POKE631,19:FORX=
      632TO640:POKEX,13:NEXT:P
      OKE198,10            :rem 94
```

©

Adding TIME$ To Atari

Kenneth S Szajda

Here's a useful routine that adds a missing feature to Atari BASIC: TIME$. Now your programs can have realtime clocks and timed loops without PEEKs or POKEs. Requires only 8K RAM for cassette or 16K RAM for disk (with DOS 2.0, 2.5, and 3.0).

Atari BASIC is a very versatile and useful language. However, like all computer languages, it is not perfect. One useful feature that Atari BASIC lacks is the TIME$ function. For beginners, TIME$ provides a method to accurately time loops; for advanced programmers, TIME$ is a useful tool for avoiding the system timers, saving a lot of extraneous coding.

Timing from BASIC usually comes in two forms: TIME$ and TI (or some other appropriate numeric variable). Both supply the same information, but TIME$ represents the time as "HH:MM:SS" (or "HHMMSS") and TI represents the time in *jiffies* (1/60 second). In general, TIME$ is more useful because it is already formatted and ready for printing. Program 1 adds the TIME$ function to Atari BASIC, giving you easy access to the time without ever touching the system timers.

A Few Rules

Since we're patching this function into Atari BASIC, there are a few rules to follow for it to work properly. First, TIME$ must be DIMensioned like any other string variable, and it *must* be the first variable of any kind to appear in your program. To DIMension it, always use DIM TIME$(8). To activate the routine, use A=USR(1536) after it has been loaded into memory. (The routine automatically changes the dimension to eight and the length to eight no matter what you specify in the DIM statement, but it is best to use the correct value to avoid slight inaccuracies rippling throughout BASIC as a result.)

Second, TIME$ must always be the first variable BASIC sees when you LOAD or ENTER a program. If a new program is typed in with TIME$ as the first variable, no problems will result. However, adding TIME$ to a program already on tape or disk is a little tricky, but not very difficult. First, load your program into the computer (using LOAD or ENTER). When the READY prompt appears, add DIM TIME$ (8): A=USR(1536) as the very first executable (non-REM or DATA) line in your program. Then add any features using TIME$ to your program. (This step can be done at any time as long as the first line is already in the program.) Then store the program on tape or disk with the LIST command.

Using SAVE instead of LIST disables TIME$ when the program is loaded, because TIME$ will not be the first variable in BASIC's *variable name table* (which holds the name of every variable in your program). If you want to store the program with SAVE rather than LIST, just type NEW (to clear the variable name table) and reENTER the program after you have LISTed it to tape or disk. This rewrites the variable name table in the order that the variables appear in the program: TIME$ should be first.

Once TIME$ is first in the variable name table, subsequent SAVEs will not change its position, and everything will work smoothly.

The last rule, and probably the most important, is to avoid the CLR command. When CLR is executed while TIME$ is activated, the time will be lost and TIME$ will show "junk." Then you'll have to either type NEW and repeat the steps above or simply rerun the program (since RUN clears all variables and cleans things up).

An Autoboot Routine

The A=USR(1536) statement after the DIM statement actually sets TIME$ to "00:00:00" and activates the routine. The time will be expressed in military format, from 00:00:00 to 23:59:59. Then it returns to 00:00:00 and begins to count upward again.

The TIME$ program was written to be used as an autoboot file—AUTORUN.SYS with disk and a boot tape with cassette. For a disk-based system, run Program 1 and specify disk at the prompt. Remember, however, that this will erase an existing AUTORUN.SYS file on the disk, unless the AUTORUN.SYS file is locked, in which case the SAVE will fail.

For cassette-based systems, run Program 1 and specify cassette. The loader program will modify the TIME$ routine to make a boot tape.

In either case, loading the program on power up is simple. For disk, just boot up with the disk containing the AUTORUN.SYS file.

The routine loads into memory and can be activated with the DIM TIME$ (8):A=USR(1536) sequence. For cassette, boot the program by holding down the START button while turning on the computer. Press RETURN when the buzzer sounds, and the program automatically loads into memory. Again, activate the routine with the DIM TIME$(8): A=USR(1536) sequence.

You can set the time by using normal string manipulations. For example, to set TIME$ to 11 a.m., just type TIME$="11:00:00". If you press SYSTEM RESET, the time will *not* be stopped. Once started, the only way to stop the time is to use NEW. As long as TIME$ remains first in the variable name table, the counter will continue to update the time. No other special commands or techniques are required.

To see TIME$ at work, try running Program 2 after TIME$ is activated. The program asks for the current time and the time to sound an alarm (both in HH:MM:SS format). The computer displays the time until the alarm time arrives and then sounds five bell characters. Program 2 shows how TIME$ can be used like any other string variable.

How TIME$ Ticks

The TIME$ routine takes advantage of the Atari's timers. Atari computers contain many timers, but the ones most often used are the five two-byte timers at memory addresses 536–545 ($218 to $221 hex). Each timer is set up in the usual 6502 least significant byte (LSB), most significant byte (MSB) order.

Unlike most of the other system timers, however, these timers count down to zero instead of counting up from zero. During each vertical blank period, each timer is decremented. Since a vertical blank occurs every 1/60 second and the highest timer value possible is 65535, the timers can time a maximum of 18 minutes, 12¼ seconds each. When a timer counts down to zero, one of two things happens: Either a flag is set or a JSR (machine language Jump to SubRoutine) is executed. For timers 3, 4, and 5, a flag is set: CDTMF3 (address 554, $22A hex) for timer 3, CDTMF4 (556, $22C hex) for timer 4, and CDTMF5 (558, $22E hex) for timer 5. When timer 1 or 2 counts down to 0, a JSR is executed through CDTMA1 (550 and 551, $226 and $227 hex) for timer 1, and CDTMA2 (552 and 553, $228 and $229 hex) for timer 2.

TIME$ uses timer 2. Timer 3, 4, or 5 would require another routine to monitor the appropriate flag. Timers 2 through 5 are stopped during critical vertical blanks, which occur during input/output with peripherals. Timer 1 seems ideal, since it's the only timer that isn't affected by the critical vertical blanks. However, the serial input/output handler (SIO) uses it as a device timeout timer (to provide us with the ever-famous ERROR-138). So the TIME$ routine is forced to use timer 2.

Furthermore, to create an interrupt every second, the routine stores a value of 60 into the timer (remember, the timer is decremented every 1/60 second), and the interrupt service routine resets the timer to 60 at each interrupt. The only drawback is that input/output with peripherals temporarily stops the timer, and TIME$ will be slightly behind the true time (but can be easily changed as shown above).

Startup Routines

The initial call to the startup routine checks to see if TIME$ is the first variable and, if so, sets up the vector for the interrupt update routine, intercepts the SYSTEM RESET initialization vector (to keep the interrupt routine going after a SYSTEM RESET), determines the address of TIME$, sets its length and dimension to eight, initializes TIME$ to 00:00:00, and returns to BASIC. The interrupt service routine again checks to make sure TIME$ is the first variable (just in case a NEW was executed), determines the address of TIME$ again (since its address may change as a result of additions and corrections to the program in memory), sets the length and dimension to eight (just in case), and resets the timer value to 60 to cause another interrupt one second later.

Of course, during this entire process, TIME$ is updated to reflect the change in time. If a NEW has been executed since the last interrupt, the value of TIME$ is *not* updated and the timer is *not* reset to 60. In other words, if a NEW occurs, the routine effectively dismantles itself. The routine must be restarted with another DIM TIME$(8):A=USR (1536) sequence.

The program traps NEW but not CLR because it is difficult to tell when a CLR has been executed—CLR does not cause any actions within BASIC's tables that could not be caused by some other command or routine. Since NEW effectively blocks the variable name table, it is relatively simple to check for.

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering these listings

Program 1: TIME$ BASIC Loader

```
BH 10 DIM TIME$(8)
IL 20 GRAPHICS 0:POKE 752,1:
      ? "PROCESSING..."
IB 30 CHECKSUM=0:RESTORE
EG 40 FOR X=1536 TO 1758:REA
      D A:CHECKSUM=CHECKSUM+
      A:POKE X,A:NEXT X
DL 50 IF CHECKSUM<>25068 THE
      N ? :? "{2 BELL}***ERR
      OR IN DATA STATEMENTS*
      **":END
JI 60 OPEN #2,4,0,"K":? :? "
      Cassette or Disk ?";:G
      ET #2,A
ON 70 IF A<>67 AND A<>68 THE
      N CLOSE #2:GOTO 60
KI 80 IF A=67 THEN 160
LL 90 ? :? "Type Y to create
       AUTORUN.SYS":? "NOTE:
       Existing AUTORUN.SYS
      will be deleted"
CO 100 GET #2,A:IF A<>89 THE
       N END
GN 110 TRAP 120:XIO 33,#1,0,
       0,"D:AUTORUN.SYS"
BM 120 TRAP 65535:OPEN #1,8,
       0,"D:AUTORUN.SYS":PUT
        #1,255:PUT #1,255:PU
       T #1,0
HF 130 PUT #1,6:PUT #1,222:P
       UT #1,6
KO 140 FOR X=1536 TO 1758:PU
       T #1,PEEK(X):NEXT X:C
       LOSE #1
CM 150 ? :? "AUTORUN.SYS is
       now on disk":POKE 752
       ,0:NEW
EN 160 ? :? "Position tape,
       press REC and PLAY":?
        "and press RETURN...
       ";:GET #2,A
KN 170 POKE 1528,0:POKE 1529
       ,2:POKE 1530,248:POKE
        1531,5:POKE 1532,255
       :POKE 1533,5
PJ 180 POKE 1534,24:POKE 153
       5,96
HH 190 POKE 1601,2:POKE 1605
       ,3:POKE 1741,234:POKE
```

```
       1742,234:POKE 1743,2
       34
BI 200 POKE 764,0:OPEN #1,8,
       128,"C:":POKE 764,255
GA 210 POKE 850,11:POKE 852,
       248:POKE 853,5:POKE 8
       56,231:POKE 857,0
ME 220 A=USR(ADR("hhh■LVE"),
       16)
AH 230 CLOSE #1:? :? "Boot f
       ile is now on tape":P
       OKE 752,0:NEW
LP 240 DATA 104,169,0,133,17
       8,160
FP 250 DATA 4,177,130,217,21
       8,6
JF 260 DATA 208,56,136,16,24
       6,32
HA 270 DATA 156,6,165,178,20
       8,9
KD 280 DATA 160,8,169,48,145
       ,176
MO 290 DATA 136,16,251,169,5
       8,160
FJ 300 DATA 2,145,176,160,5,
       145
FN 310 DATA 176,169,71,141,4
       0,2
CO 320 DATA 169,6,141,41,2,1
       69
OJ 330 DATA 2,162,0,160,60,3
       2
NA 340 DATA 92,228,169,198,1
       33,12
GF 350 DATA 169,6,133,13,96,
       160
GB 360 DATA 4,177,130,217,21
       8,6
MN 370 DATA 208,66,136,16,24
       6,216
GM 380 DATA 32,156,6,169,58,
       160
GC 390 DATA 2,145,176,160,5,
       145
JJ 400 DATA 176,160,7,177,17
       6,24
LH 410 DATA 105,1,145,176,21
       7,210
GI 420 DATA 6,144,25,169,48,
       145
PO 430 DATA 176,136,177,176,
       24,105
FK 440 DATA 1,145,176,217,21
       0,6
KE 450 DATA 144,8,169,48,145
       ,176
JJ 460 DATA 136,136,16,219,1
       69,2
CH 470 DATA 162,0,160,60,32,
       92
DI 480 DATA 228,96,169,2,162
       ,0
DF 490 DATA 160,0,32,92,228,
       96
MA 500 DATA 160,0,169,129,14
       5,134
GB 510 DATA 160,4,169,8,145,
       134
DM 520 DATA 200,200,145,134,
       136,169
KO 530 DATA 0,145,134,200,20
       0,145
IN 540 DATA 134,160,2,177,13
       4,24
BN 550 DATA 101,140,133,176,
       200,177
PB 560 DATA 134,101,141,133,
       177,96
MC 570 DATA 169,255,133,178,
       32,17
MA 580 DATA 6,32,64,21,24,96
AP 590 DATA 51,52,48,54,58,4
       8
BF 600 DATA 54,58,84,73,77,6
       9
EM 610 DATA 164
```

Program 2: Sample TIME$ Program

```
AD 10 DIM TIME$(8),A$(8):A=U
      SR(1536)
OD 20 GRAPHICS 0
OO 30 ? "Enter current time
      ";:INPUT TIME$
JA 40 ? :? "Enter time to so
      und alarm ";:INPUT A$
OH 50 GRAPHICS 1
EL 60 POSITION 5,9:? #6;TIME
      $:POKE 708,INT(PEEK(53
      770)/16)*16+8
ON 70 IF TIME$=A$ THEN POSIT
      ION 5,9:? #6;TIME$:? "
      {5 BELL}{CLEAR}It's ti
      me...":POKE 708,40:END
AH 80 GOTO 60
```

©

Apple Program Protector

Boris Troyanovsky

Do you have an Applesoft BASIC program you want to protect from prying eyes? With this technique, you can keep other people from listing your programs—while still giving them the freedom to make copies. For all Apple II-series computers with DOS 3.3.

"Apple Program Protector" is an easy to use utility that keeps other people from listing your BASIC and machine language programs. It works by moving the disk catalog on the protected disk to another track, and by preventing users from breaking out of the program by pressing CTRL-C or RESET. The only requirement is that your program must run itself when the user boots the disk.

It's easy enough to ensure that the program runs automatically when the disk is booted. DOS 3.3 always loads and runs a BASIC program whenever a disk is booted if the program is named HELLO. Only a single BASIC program can be started in this way. If you wish to have more than one BASIC program on the protected disk, you must make HELLO a menu program that allows you to select the desired program from the disk. If the program you wish to protect is written in machine language, you can use HELLO to start it. For example, if you want to protect a program named MLGAME, your HELLO program might be simply:

```
10 PRINT CHR$(4);"BRUN MLGAME"
```

The next step is defend against CTRL-RESET and CTRL-C. In each BASIC program on the disk you wish to protect, add these two lines:

```
0 POKE 1011,0:ONERR GOTO 63999
63999 RESUME
```

The POKE in line 0 defends against the CTRL-RESET key (or just RESET on some Apples). If the CTRL-RESET key is pressed, any Applesoft BASIC program in memory is erased and the computer reboots.

Since the CTRL-C interrupt code (which is used to stop program execution) is considered an error by Applesoft BASIC, the ONERR GOTO statement in line 0 transfers program control to line 63999 when CTRL-C is encountered. If your program uses ONERR to test for other conditions, include the statement IF PEEK(222)=255 THEN 63999 in your error testing. This ensures that CTRL-C is still trapped.

Machine language programs can be protected in this way, too. Just add these lines to your source code:

```
LDA #$00
STA $03F3
```

Like the BASIC lines above, these instructions erase the program from memory and reboot the computer if CTRL-RESET is pressed.

Preparing A Protected Disk

Begin by preparing a disk containing the program or programs you wish to protect. If the disk is to contain only one program, load it into memory, insert a new disk, type INIT HELLO, and hit RETURN. If you wish to have several programs on the disk, the HELLO program must be a menu program and the others can simply be saved on disk in the usual manner. Remember to add to each program the lines mentioned above to protect against CTRL-RESET and CTRL-C. *Make sure all programs are fully tested and debugged before you run Program Protector.* To be safe, you should always keep backup copies of the programs on an unprotected disk.

Next, type in Program 1 below and save a copy on a separate disk. Use the filename PROTECTOR. With the built-in machine language monitor, enter the data from Program 2. (If you are unsure about using the monitor, consult your user's manual.) Save the machine language onto the same disk with Program 1 using the command BSAVE IOB, A$0300, L$40.

The Program Protector disk is now ready to use. To protect one of your disks against intrusion by outsiders, follow these steps:

1. Insert the Program Protector disk into the drive (the drive should be addressed as slot 6, drive 1).
2. Type BLOAD IOB.
3. Type RUN PROTECTOR.
4. Program Protector is now loaded into memory. You should see the prompt DESTINATION TRACK: on the screen. Remove the disk with Program Protector and insert the disk that you'd like to protect into drive 1. Type the number of the track to which you want to move the catalog and press RETURN. The number must be greater than or equal to 3 (DOS occupies tracks 0–2), and less than or equal to 34, since there are only 35 tracks (numbered 0–34) on the disk. Also, the number can't be 17, because that's where the catalog is already.
5. The disk drive whirs a bit, then the Applesoft] prompt reappears. The disk is now protected. If you type CATALOG, you'll be shown an empty directory. If you try to load a program from the protected disk or save an additional program to it, you'll get nothing but a DISK FULL error message. However, if you now boot the protected disk, the HELLO program loads and runs normally, except that CTRL-C no longer stops the program and CTRL-RESET only reboots the system.

There is a way to regain access to the programs on the protected disk. Boot a normal disk, then enter POKE 44033,*n* (substitute for *n* the number of the track to which the catalog was moved). You can now display the catalog and load and save programs. You can also use this technique if the program you're protecting needs to access another program on a different disk. To let the program know where the catalog of the new disk is, POKE 44033 with the catalog track of the disk you'd like to access.

Don't try to relocate the catalog on a disk more than once. The results are unpredictable.

Program 1: Apple Program Protector

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!

```
7B 1Ø HIMEM: 8191
E8 2Ø HOME : HTAB 11: INVERSE :
   PRINT "PROGRAM PROTECTOR":
    NORMAL : VTAB 1Ø: INPUT "
   DESTINATION TRACK: ";DT
ED 3Ø IF DT < 3 OR DT > 34 OR DT
    = 17 THEN PRINT "ILLEGAL
   TRACK NUMBER": FOR A = Ø T
   O 6ØØ: NEXT A: RUN
9E 4Ø REM **MOVE TRACKS**
31 5Ø SS = Ø:SE = 15:TR = 17:BU
   = B192:OP = 1: GOSUB 38Ø:
   REM READ CATALOG TRACK
51 6Ø GOSUB 18Ø
AA 7Ø D1 = PEEK (DT * 4 + 56 + 8
   192):D2 = PEEK (DT * 4 + 5
   7 + B192):O1 = PEEK (17 *
   4 + 56 + B192):O2 = PEEK (
   17 * 4 + 57 + B192)
DB 8Ø POKE 17 * 4 + 56 + 8192,D1
   : POKE 17 * 4 + 57 + 8192,
   D2: POKE DT * 4 + 56 + 819
   2,O1: POKE DT * 4 + 57 + B
   192,O2
EA 9Ø SS = Ø:SE = 15:TR = DT:BU
    = 163B4:OP = 1: GOSUB 3BØ:
    REM READ NEW TRACK
64 1ØØ REM **ALTPOINT ROUTINE**
C9 11Ø TV = B192
56 12Ø FOR TX = Ø TO 15: POKE TV
    + 1,DT:TV = TV + 256: NE
   XT TX
CD 13Ø SS = Ø:SE = 15:TR = DT:BU
    = B192:OP = 2: GOSUB 38Ø
   : REM WRITE CATALOG TRACK
5D 14Ø SS = Ø:SE = 15:TR = 17:BU
    = 16384:OP = 2: GOSUB 3B
   Ø: REM WRITE NEW TRACK
F8 15Ø SS = 11:SE = 11:TR = 1:BU
    = 8192:OP = 1: GOSUB 38Ø
   : POKE B193,DT:SS = 11:SE
    = 11:TR = 1:BU = B192:OP
    = 2: GOSUB 3BØ: REM CHAN
   GE DOS
4B 16Ø GOSUB 23Ø
97 17Ø END
04 18Ø REM **ALTER CATALOG'S T/S
    POINTERS**
BE 19Ø BF = B192 + 256
DD 2ØØ FOR PR = 11 TO 221 STEP 3
   5: IF PEEK (BF + PR) = DT
    THEN POKE BF + PR,17
3B 21Ø NEXT PR:BF = BF + 256: IF
    8F < > 122B8 THEN GOTO 2
   ØØ
16 22Ø RETURN
99 23Ø REM **CHANGE T/S LISTS**
B5 24Ø BF = B192 + 256
CC 25Ø FOR PR = 11 TO 221 STEP 3
   5: IF PEEK (BF + PR) < >
   Ø THEN GOSUB 29Ø
41 26Ø NEXT PR
B6 27Ø BF = BF + 256: IF BF < >
   122BB THEN GOTO 25Ø
22 2BØ RETURN
A1 29Ø REM **DIRTY WORK**
ED 3ØØ LT = PEEK (BF + PR):LS =
   PEEK (BF + PR + 1)
AB 31Ø SS = LS:SE = LS:TR = LT:B
   U = 163B4:OP = 1: GOSUB 3
   BØ:BU = BU - 256
9C 32Ø FOR CT = 12 TO 254 STEP 2
   : IF PEEK (BU + CT) = DT
   THEN POKE BU + CT,17
B7 33Ø NEXT CT
B9 34Ø IF PEEK (BU + 1) = DT THE
   N POKE BF + 1,17
3B 35Ø OP = 2:SS = LS:SE = LS:TR
    = LT:BU = 163B4: GOSUB 3
   BØ:BU = 8U - 256
53 36Ø IF PEEK (BU + 1) < > Ø TH
   EN LT = PEEK (BU + 1):LS
   = PEEK (BU + 2): GOTO 31Ø
21 37Ø RETURN
7E 38Ø REM **DISK ACCESS**
91 39Ø FOR SA = SS TO SE
FB 4ØØ POKE 7BB,TR: POKE 7B9,SA:
    POKE 796,OP
CB 41Ø HB = INT (BU / 256):L8 =
   8U - (HB * 256)
67 42Ø POKE 792,L8: POKE 793,HB
28 43Ø CALL 76B:8U = 8U + 256: N
   EXT SA
1C 44Ø RETURN
```

Program 2: IOB Routine

Enter this listing with the machine language monitor

```
Ø3ØØ- A9 Ø3 AØ 1Ø 2Ø D9 Ø3 6Ø
Ø3ØB- ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
Ø31Ø- Ø1 6Ø Ø1 ØØ 11 ØF 3Ø Ø3
Ø318- ØØ 2Ø ØØ ØØ Ø1 ØØ FE 6Ø
Ø32Ø- Ø1 ØØ ØØ ØØ ØØ ØØ ØØ ØØ
Ø328- ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
Ø33Ø- ØØ Ø1 EF D8 ØØ ØØ ØØ ØØ
Ø338- ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
```

©

The Beginners Page

Tom R. Halfhill, Editor

Variable Accuracy

There are a few more points about integer variables that we didn't cover last month—including some important exceptions to general rules.

Note that in some versions of BASIC, such as Commodore BASIC, you aren't allowed to use integer variables as counters in FOR-NEXT loops. In other words, a statement such as FOR X%=1 to 10:NEXT X% would cause an error. However, integer variables *can* be used as counters in IBM BASIC. Just remember that because integer variables are restricted to a minimum value of −32,768 and a maximum of 32,767, you'll have to make sure your loops don't exceed those limits.

In some BASICs, there's also one exception to the rule about denoting all integer variables with the % symbol. In IBM BASIC, for instance, you can insert a DEFINT (*define integer*) statement near the beginning of the program to define a whole group of variable names as integer variables by default. The statement DEFINT A–M declares that all variable names beginning with the letters A through M are automatically integer variables. Since integer variables execute faster and consume less memory on the IBM than regular variables, a DEFINT statement can improve a program's performance. (Remember from last month that integer variables don't save memory and actually run slower on Commodore and Apple computers.)

Incidentally, another way to speed up your programs is to replace frequently used constants with variables. In most Microsoft BASICs, variables execute faster than constants (the reverse is true of Atari BASIC). By predefining the most commonly used numbers (usually 0–10) as variables, statements such as Y=Y+1 can be changed to Y=Y+C1. And if you're programming on an IBM, you can define them as integer variables and pick up even more speed. Try this technique in your next program and see if it adds a little zip.

Improving Precision

Another type of variable is provided in some BASICs to improve mathematical accuracy. Called *double-precision* variables, they can help avoid the small rounding errors that sometimes accumulate and cause strange results. (Rounding errors are slight discrepancies that can crop up when the computer converts our everyday decimal numbers to its internal system of binary numbers, performs some arithmetic, and then converts the answer back into decimal again.)

Double-precision variables are available in IBM BASIC and some versions of TRS-80 BASIC, but not in Commodore BASIC, Applesoft, Atari BASIC, or TI BASIC. Even if your BASIC doesn't have double-precision variables, however, they're worth learning about. As personal computers grow more powerful, you're sure to encounter such features sooner or later.

Here's an example which demonstrates a common type of rounding error—in this case, on the IBM PC/PCjr. This program initializes the variable Y to 100, then subtracts the value .05 ten times using a FOR-NEXT loop. The final value of Y is printed after the loop is finished.

```
10 Y=100
20 FOR X=1 TO 10
30 Y=Y-.05
40 NEXT X
50 PRINT Y
```

The answer, of course, should be 99.5. Instead, here's the program's answer:

```
99.49997
```

If you swap the statements in lines 40 and 50 so the program prints the current value of Y after each pass through the loop, you'll see that the rounding error starts with the second calculation and keeps increasing until the final result is off by .00003. That's not a huge discrepancy—but still, if this were some kind of banking program that was subtracting a nickel-a-day service charge from customer accounts over a period of time, someone might be cheated out of a penny now and then. (It's happened to me, by the way.)

Fortunately, you can program the computer to deliver a better answer. To convert Y from a regular (single-precision) variable into a double-precision variable, add the # symbol to every occurrence of Y in lines 10, 30, and 50. The modified program yields this result:

```
99.49999999254942
```

Well, even computers aren't perfect. This time the rounding error starts with the first calculation, although the errors are smaller and the final answer is off by only .0000001192093. For even greater accuracy, we can define the value of .05 as a *double-precision constant* by adding the # symbol to .05 in line 30. Here's the result:

```
99.49999999999999
```

This is even more accurate; now the computer is off by only .00000000000001. Furthermore, by switching lines 40 and 50 to see the results of each calculation through the loop, you'll notice that all the intermediate answers are exactly correct. That still leaves us with an infinitesimal error in the final answer, however. If you're a stickler for absolute accuracy, there are ways to get around these runaway fractions—but that's a topic for a future column. ©

David D. Thornburg, Associate Editor

Personal Computers And Personal Freedom

Last July I spent a day in Colonial Williamsburg, Virginia, catching a glimpse of life as it was in the mid-1700s. The publishing exhibit particularly caught my attention. One could see paper being made by hand—a process that required skill and strength on the part of the craftsman, and which took a long time. In the print shop one could see the pages of a book being printed from hand-set type—another expensive and time-consuming process. The bindery exhibit showed how the printed pages were folded into signatures and stitched together by hand before being bound in leather.

It was clear from this exhibit that access to books was limited to the wealthy. The cost of spreading the printed word was quite high, and yet this period gave us a rich collection of people who had much of importance to say—Patrick Henry and Thomas Paine, to name just two.

As I thought about our advances in communication technology since that period—typewriters, copiers, computers, and so on—it became clear that the reduced cost of communication was one of the main reasons that literacy could spread to the public at large. The printed word has spread like wildfire, carrying messages into homes that would have been bookless in the 1700s.

The freedom to communicate is one of our most treasured freedoms. There are nations on this planet where individual ownership of copiers and computers is forbidden. It is easy to see why—it's important for a totalitarian government to control the flow and distribution of information. Otherwise, individuals could create, publish, and distribute their own ideas without the censorship of the state.

Computer Publishing

Prior to the widespread sale of personal computers, we had restrictions of our own that limited the widespread dissemination of ideas.

Before an opinion can be expressed in printed form, the author must either convince a publisher that it is worth expressing, or must elect to publish it alone. Even if a publisher accepts a work, it will reach an audience only if stores decide to stock it.

Suppose you've written something you think others might like to read—a collection of poetry, for example, or a political treatise. You may find that traditional publishers are not interested in your material because your market is too specialized. Or, you may find that they are interested, but that if you wait the four to six months (or longer) that it takes for your words to be printed, your material will have lost its currency and impact.

In this case, you may elect to publish the material yourself.

Prior to the personal computer, you might be restricted to running copies of your material at the local print shop. Depending on the size of your document, you may find that it costs several dollars per copy to have it printed.

But, in an era where personal computers are increasingly commonplace, there's another way of publishing your ideas—especially if what you have to say is of particular value to others who own computers. You can publish your ideas on a disk! Disks are inexpensive, reusable, and can be duplicated as needed. Publishing your material on disk lets you fix mistakes quickly without having to wait for a new printing. Your material might consist of text files that can be read with a word processor, or you can write your own program that lets people read or print your files as they choose.

However, along with the freedom to publish your own materials in the privacy of your home using nothing more than your personal computer there comes a responsibility. It is correctly said that the pen is mightier than the sword. As your own publisher, you can say anything you wish, but you must always keep in mind that the printed (or displayed) word is very powerful. Think your ideas through carefully before publishing them.

Talking Books

I recently used this publication technique for my book *In Search of the One-Minute Megatrends—Surviving the Bad Times in Silicon Valley* (Innovision Press, $12.95). While the information in this book is of potential interest to a broader audience, I initially made it available on a Macintosh disk, since that's the computer with which I do most of my writing. As I was creating the book, it occurred to me that this method of publishing had much greater flexibility than the printed page. For example, readers could change the typeface and size if they desired.

I also included a set of files on the disk that lets the book read itself aloud to the user with the Smooth-Talker speech synthesizer from First Byte. This not only provides another alternative for reading the book, but also makes the material available to those with impaired vision.

While this book is not available in stores, I have been able to sell it quite well through direct mail by placing inexpensive advertisements in regional computer-interest newspapers. The success of this venture convinces me that anyone with a message of interest to computer owners can be an author and a publisher as well.

If only the founding fathers could see us now! ©

Arlan R Levitan

Faster Than A Speeding Byte

Last month I mentioned Fastlink, a new 10,000 bits per second modem from Digital Communications Associates. Not only is it five times faster than the latest "high-speed" 2400 bps modems, it even works over regular telephone lines. Until now, anything close to 10,000 bps required you to lease special data-grade lines from the phone company. But not the DCA Fastlink. Although the Fastlink's $2,000 price tag is a little rich for most casual users' blood, there has been a fair amount of incredulous reaction like "how the heck can they do 10,000 bits per second?" from readers of this column (and even from my editor).

While the subject is a bit technical, I've distilled an explanation that will either satisfy your curiosity or teach you never to ask me about this sort of thing again. To get started, let's review our old friend, the ordinary 300 bps modem.

Modems exchange information over phone lines by transmitting and receiving audio tones. A 300 bps modem transmits over two channels, one for each direction. Each second of time is divided into 300 slices, and each slice is called a *baud*. A 300 bps modem packs one bit into each baud (1200 and 2400 bps modems both operate at 600 baud and pack 2 and 4 bits into each baud, respectively). One channel transmits signals in the audio range of 1070–1270 hertz, and the other at 2025–2225 hertz. That means each channel has a fairly wide bandwidth (200 hertz), and they're separated by a guard band of no signal (755 hertz wide) that makes it easy for the modem circuitry to differentiate between the two channels.

DCA's Fastlink uses a very low 7.3 baud rate, so it can drastically narrow the channel bandwiths and guard bands. The Fastlink also uses the entire 0–4000 hertz audio spectrum of normal phone lines. When two Fastlink modems link up, they attempt to establish a maximum of 512 separate channels, each 7.8 hertz apart. They analyze each channel to determine which ones are noise-free enough to handle transmission techniques that pack 4 or 6 bits into each time slice, or baud.

Then the Fastlink transmits data by using a hybrid parallel/-serial system (300 bps modems send data in a serial stream of bits—one bit after another with one bit per baud). The bits carried by all channels in use during one baud are considered a single packet of information. Outgoing data bits are assigned to channels as they're prepared for sending (with either 4 or 6 bits per channel), beginning with the channels at the lowest frequencies. Once the packet is assembled, it's sent across the active channels. So the data is sent in parallel within the packets, and the packets themselves are sent serially.

Blistering Speed

Using the Fastlink method, the maximum theoretical throughput is 512 channels × 6 bits per baud × 7.3 baud per second, or more than 20,000 bps. Given the quality of most voice-grade lines, that limit is very theoretical. Most channels operate at only 4 bits per baud, and throughput is further limited by the overhead of error detection and correction, which is automatically handled by Fastlink. All these factors reduce the Fastlink's actual throughput to a blistering 10,000 bps on local phone lines. A Fastlink modem operating on lines provided by the most popular long-distance carrier should work at about 8,000 bps. On the lines provided by other common carriers, the Fastlink averages about 7,000 bps.

The Fastlink monitors the quality of the phone line during the linkup, shutting down channels that become marginally acceptable or opening up channels if quality improves. DCA refers to the process as DAMQAM, or *Dynamically Adaptive Multicarrier Quadrature Amplitude Modulation* (say it five times fast). To handle all this data manipulation and line monitoring, the Fastlink is actually a full-fledged, highly specialized computer with a megabit of memory and two central processing units—a Motorola 68008 working in tandem with a Texas Instruments 320.

There are some fine points to keep in mind while daydreaming about cruising along at 10,000 bps. The Fastlink dynamically assigns channels to incoming or outgoing data based on the volume going back and forth. If there is an equal amount of data moving in both directions, the Fastlink channels would be equally divided between incoming and outgoing data, resulting in an effective maximum speed of only 5,000 bps for each data stream.

In practice, the data flow is usually quite lopsided, with ratios of 99 to 1 more common than 50:50. So the bulk of data flow on a Fastlink is assigned the lion's share of channels, resulting in throughput that is very close to the 10,000 bps ideal.

Fastlink modems currently come in two flavors. An internal version for the IBM PC and compatibles goes for $1,995 and includes a special version of Microstuf's *Crosstalk* program adapted for the Fastlink. An outboard RS-232 Fastlink is priced at $2,395. Both modems are also capable of communicating at plain old 300 or 1200 bps with non-Fastlink modems. If you're still curious, you can get even more information by contacting DCA at 1000 Alderman Drive, Alpharetta, GA 30201. ©

The World Inside the Computer

Fred D'Ignazio, Associate Editor

The Case Of The Phantom Programmers

Earlier this year I wrote about one of my high school assistants—Howard Boggess, my "Computer Handyman." This time I'd like to introduce you to another one of my assistants—Hunter Baker, my "Phantom Programmer."

Like Howard, Hunter came to me from David James's computer science class at Patrick Henry High School, here in Roanoke, Virginia. When Hunter arrived at my house on the first day, I took him and his mother to the dark, hot attic where Howard had rescued several broken-down computers (see "The World Inside The Computer," COMPUTE!, January 1985). "This is your first task," I said, with a sweep of my arm. "If you can clean this attic, then I know you can do anything."

Hunter is a quiet, mild-mannered person. He simply nodded when I told him to clean the attic. But this was no ordinary attic. And I worried about him every day when he trudged up the attic stairs.

I shouldn't have worried. Sending Hunter into the attic was like sending Cinderella into her stepmother's kitchen, or Hercules into the Augean stables. In a month, Hunter had the attic better organized than the rest of the house. He had everything filed away in labeled filing cabinets and had built a computer database so we could instantly know where to look for our long underwear, computer manuals, extension cords, extra paper, Christmas tree lights, winter gloves and mittens, and RS-232 cables.

Then Hunter moved downstairs. When he first confronted the downstairs office, computer software was piled to the ceiling and computer cables and circuit cards spilled out the door into the middle of the living room. But, for Hunter, after facing the horrors of the attic, this awful mess was no more than a tasty dessert. In only a couple of weeks everything was cataloged, labeled, and filed. The mess had vanished, and Hunter was hard at work at one of the computers.

Computer Trivia

One day I walked into the room, looked over Hunter's shoulder at a BASIC program on the display screen, and asked him what he was doing. He explained that he and his friend Amy Powell were doing a computer project for National History Day. They planned to create a history trivia game on the IBM computer, and Hunter asked if he and Amy could start coming over to our house after hours to work on the program. "Of course," I said, since I was sure he was only talking about a couple of evenings and maybe a weekend or two.

Ha! After watching Hunter clean the attic and the office, I should have been wiser. Hunter doesn't do anything halfway, and this project was no exception. For the next month, he and Amy came over almost every night after dinner, and most Saturdays and Sundays. They rarely left until the wee hours of the morning.

One night I was awakened around 2 a.m. by strange clicking noises. Alarmed, I tiptoed to the bedroom closet and grabbed the machete my parents had bought me in the Dominican Republic. (The machete was duller than a letter opener, and it had a parrot inscribed on its side, but it looks deadly, especially when I wave it threateningly above my head.)

I made my way cautiously down the stairs. I noticed a light was switched on in the downstairs office. I guessed that a thief must be inside stealing one of my beloved computers!

Leaping down the remaining stairs, I burst into the office, screaming and waving the machete.

It took a moment for my eyes to adjust to the bright lights in the room. When they did, I noticed Hunter and Amy seated at two IBM computers, working on their History Day program. "We're sorry we're here so late," said Hunter politely.

"Tomorrow's the competition," explained Amy. ©

INSIGHT: Atari

Bill Wilkinson

Deactivating BASIC

My coworkers and I have received many requests from owners of the Atari 600XL, 800XL, and 130XE for a simple way to turn off the BASIC built into those computers. Of course, the method recommended by Atari is to hold down the OPTION button when you boot the system. If you forget to do this when booting a program that doesn't require BASIC, the ROM-based BASIC occupies address space that costs you more than 8,000 bytes of RAM. There are other reasons for turning off BASIC as well. For instance, you might like to turn it off temporarily to gain extra memory while duplicating a few files or disks. These jobs take less time and fewer disk swaps if the computer can use the 8K of memory vacated by disabling BASIC. And avoiding a reboot or two can save time, too.

Our solution is a pair of short machine language programs that let you turn BASIC on and off from DOS. (Note that they can't turn off a BASIC *cartridge*—or any other cartridge, for that matter—so they serve no purpose on the Atari 400, 800, and 1200XL computers.) Atari manuals suggest that turning off the built-in BASIC is as simple as changing one bit in the XL/XE memory control location (which used to control joystick ports 3 and 4 in the 400 and 800). That may be true if you're writing a machine language program that takes over complete control of the computer, but in many cases it doesn't work.

First, whenever you press the RESET button, the operating system restores the built-in BASIC to the state in which you booted it. Second, if you're using ordinary graphics mode screens (without a custom display list, etc.), the screen handler doesn't use the memory freed by removing BASIC. It thinks you're still using a 40K machine.

Going the other way—turning on BASIC after booting without it—can be even messier. If you suddenly enable BASIC without doing something about the screen, you'll find yourself staring at garbage as BASIC blithely wipes out the display list, screen memory, and perhaps more. Fortunately, all of these problems can be solved by following these few steps:

1. Turn the built-in BASIC off or on.
2. Tell the operating system you did so.
3. Change the master top-of-RAM pointer.
4. Close channel 0, the screen editor.
5. Reopen channel 0.

We can tell the operating system we changed the state of BASIC via the flag in memory location 1016 ($3F8). The master top-of-RAM pointer is RAMTOP at location 106 ($6A). Channel 0 is closed and reopened to force the screen driver to use the highest available memory. Don't worry if that sounds a bit arcane. The program listed here automatically creates two machine language programs that do all the work for you. Be sure to save a copy before you run it.

```
6D 100 DIM NAME$(20)
NN 110 LINE=800:GOSUB 210
NO 120 LINE=900:GOSUB 210
GL 130 END
DE 210 CHECK=0:RESTORE LINE
GH 220 FOR CNT=1 TO 57:READ
       BYTE
EL 230 CHECK=CHECK+BYTE:NEXT
        CNT
EI 240 READ TEST:IF CHECK<>T
       EST THEN STOP
OJ 250 READ NAME$:OPEN #1,8,
       0,NAME$
DE 260 RESTORE LINE
GM 270 FOR CNT=1 TO 57:READ
       BYTE
KF 280 PUT #1,BYTE:NEXT CNT
GF 290 CLOSE #1
HD 300 RETURN
IN 810 DATA 255,255,0,4,44,4
       ,173,1,211,9,2,141,1
OL 830 DATA 211,169,1,141,24
       8,3,169,12,32,24,4
DG 840 DATA 169,192,133,106,
       169,3,141,66,3,169,42
ID 860 DATA 141,68,3,169,4,1
       41,69,3,162,0,76,86
DK 870 DATA 228,69,58,0,226,
       2,227,2,0,4
HM 880 DATA 5045,D:BASICOFF.
       COM
MF 910 DATA 255,255,0,4,44,4
       ,173,1,211,41,253,141
JI 930 DATA 1,211,169,0,141,
       248,3,169,12,32,24,4
DC 940 DATA 169,160,133,106,
       169,3,141,66,3,169,42
ID 950 DATA 141,68,3,169,4,1
       41,69,3,162,0,76,86
OL 970 DATA 228,69,58,0,226,
       2,227,2,0,4
EG 980 DATA 5295,D:BASICON.C
       OM
```

The program writes two binary files to disk on drive 1, naming them BASICON.COM and BASICOFF.COM. The first turns BASIC on and the second turns it off. To use either of them from DOS, simply choose the L (load binary file) option and enter the filename when prompted. (OS/A+ and DOS XL users need only type BASICON or BASICOFF in response to the D1: prompt.)

The next time you need to duplicate a disk or large file, load BASICOFF.COM first, copy the disk or file, then load BASICON.COM to reactivate BASIC. You'll save time, especially on a single-drive system. If you're writing machine language programs, call BASICOFF as a subroutine when you start your program. ©

Donald B Trivette

A Promise Of Things To Come

When I saw the advertisement for the Key Tronic KB 5152V, I knew it was a product designed with me in mind. Who hasn't dreamed of using a typewriter that will type every word you speak—or better yet, a computer that can understand spoken commands? The KB 5152V speech-recognition keyboard for the IBM PC, manufactured by Key Tronic of Spokane, Washington, seemed to hold just that promise. While waiting for a demonstration unit to arrive, I had visions of a new, laid-back life. Since my hands would no longer be needed for typing, I could dictate prose while holding a beverage and munching pretzels. Nor would I be restricted to a sitting position. This very column—in the interest of evaluating the product, of course—would be written from my bed.

The new keyboard arrived and plugged right into the socket vacated by the original IBM keyboard. It's an enhanced keyboard with separate numeric keypad and LED indicator lights on the Caps Lock and Num Lock keys. And, of course, there's one other enhancement: A telephone operator's headset that plugs into the back of the keyboard. Without bothering to read the manual, I spoke: "Now type this." Nothing happened, nor had I really thought it would.

The first step to using the keyboard is to teach it a vocabulary. Key Tronic supplies a menu-driven BASIC program that creates a standard ASCII text file—the vocabulary. For example, the vocabulary entry for the color BLUE might appear as BLUE;BLUE. The word to the left of the semicolon is the prompt—the word you speak; the word to the right of the semicolon is what is sent to the PC, just as though it had been typed on the keyboard. It's called the response. Thus, saying "blue" types *BLUE*. But that doesn't have to be true. You can teach the keyboard that blue is red and red is white, and it won't be the wiser.

The response characters can be more than one word and may contain characters in braces to represent keys, such as Enter, Backspace, and the special function keys. You can also define responses by keyboard scan codes or ASCII codes, so every key and key combination is accessible.

Once the written vocabulary is defined, the keyboard must be taught to recognize each word—or, more accurately, how the user pronounces each word. This is accomplished with a training session using the same BASIC program. As the computer displays each word from the vocabulary, you pronounce it at least three times. Of course, the keyboard doesn't know whether the pronunciation is correct—it can't even distinguish English from Greek or Chinese. It merely associates your pronunciation with the vocabulary word.

Voiceprinting

How does this work? The keyboard converts the sound into a pattern of zeros and ones called a *voiceprint*. As you speak, the keyboard tries to match what it hears with a previously recorded voiceprint stored in its memory. If it finds a match, the keyboard sends the appropriate word to the PC, just as though the word had been typed.

Voiceprints are stored on disk so you won't have to retrain the keyboard each morning, and the keyboard lets you mix spoken and typed input.

Following this procedure, I trained my keyboard for six words and said, "Now type this." The screen remained blank.

The manual advises, "Your voiceprints in the morning are slightly different from your voiceprints in the afternoon. Therefore you can train a vocabulary in the morning, then in the afternoon update it a few passes to build some variation into the voiceprints." With that in mind, I built variation into my vocabulary, and tried again: "Now type this." The keyboard responded by typing *this* on the computer screen.

On the subject of recognition, the manual continues, "First-time users of speech recognition products usually have poor recognition for the first few days. After working with the equipment, the ability to achieve good recognition improves dramatically. The reason for this improvement is learning to relax." When I relaxed and spoke more slowly (and stopped eating pretzels and drinking beer), the keyboard performed beautifully: *type this now type blue now type this blue*. But there's only so much that can be written with six or even 160 words. And 160 words is the vocabulary limit.

Of course, the keyboard has more serious uses than accommodating a lazy writer. As a relatively inexpensive ($995) speech-recognition product for the IBM PC, it has both industrial and personal applications. Voice recognition can be a big help to the physically handicapped. One of Boeing Computer Systems' sharpest programmers is a quadraplegic who uses a workstation built around an IBM PC-XT and a Key Tronic speech-recognition keyboard. He writes programs in BASIC and Pascal and develops spreadsheets using *Lotus 1-2-3*.

Voice recognition for the IBM PC is not advanced enough that I could comfortably write this column from a horizontal position—but surely the Key Tronic keyboard is a promise of things to come. ©

C Regena

The States Of The Union

I have previously written programs to identify the states and their capitals in certain regions ("Southern States," COMPUTE!, August 1984; "Western States" and "New England States," *Programmer's Reference Guide to the TI-99/4A*, COMPUTE! Books). Now, after numerous requests, I'll demonstrate a program for the middle Atlantic states.

The central portion of the United States can be outlined on a map as I did with the Southern States and Western States programs. However, some of the middle Atlantic states such as Delaware, Maryland, and New Jersey are too small to outline adequately on the TI-99/4A. The color-blinking method used in the New England States program won't work, either, because more color sets are needed than are available. Therefore, I've used a slightly different method for the region which includes Delaware, Maryland, New Jersey, New York, Pennsylvania, Virginia, and West Virginia. The states are drawn in high-resolution graphics with some color sets used for more than one state.

When a question mark appears on a state, the student types in the state's name. If the response is correct, the program asks for the state capital.

Be careful typing the DATA statements—make sure the commas are placed correctly. Lines 290–380 contain definitions for graphics characters. Lines 820–840 contain sets of row, column, and character numbers for drawing the graphics.

If you wish to save typing effort, you can obtain a copy of this program by sending a blank cassette or disk, a stamped, self-addressed mailer, and $3 to:

C. Regena
P.O. Box 1502
Cedar City, UT 84720

Please be sure to specify the title ("Middle Atlantic States") and that you need the TI version.

Middle Atlantic States

```
100 REM  NDRTH ATLANTIC
110 CALL CLEAR
120 PRINT "  **************
    *************":"  *";TA
    B(25);"*"
130 PRINT "  * IDENTIFY T
    HE STATES *":"  *";TA
    B(25);"*"
140 PRINT "  **************
    *************"
150 PRINT ::"{3 SPACES}NO
    RTH ATLANTIC STATES"
160 PRINT :::"TYPE THE NA
    ME OF THE STATE  THEN
     PRESS <ENTER>."
170 PRINT :"IF THE STATE
    IS CORRECT,
    {4 SPACES}TYPE THE CA
    PITAL CITY."
180 PRINT :"NAMES MUST BE
     SPELLED{7 SPACES}COR
    RECTLY TO BE ACCEPTED
    ."
190 PRINT :"PRESS FCTN 8
    REDO TO ERASE."
200 CALL CHAR(33,"3F3F3F3
    F3F3F3F3F")
210 CALL CHAR(35,"FFFFFFF
    FFFFFFFFF")
220 CALL CHAR(43,"F8F8F8F
    8F0F0F0F")
230 CALL CHAR(45,"F0F0F0F
    8F8FCFCFE")
240 CALL CHAR(47,"FFFFFFF
    FFFFFFFFF")
250 FDR C=96 TO 159
260 READ C$
270 CALL CHAR(C,C$)
280 NEXT C
290 DATA FFFFFEFEFCFCFCF8
    ,F0F0F0F0F0F8FEFF,FFF
    FFEFCFCFCF8F8,3F3F3F3
    F3F3F3F3F,0
300 DATA 0,,,000001030 73F
    FFFF,0303070 7676FFFFF
310 DATA 000001010303 0F0F
    ,0F0F0F0F0F0703,C0C
    0808080 80E0E0,F8F8F8F
    0F0E0C08,FFFFFFFEFCFC
    F8F
320 DATA F0E0602,80F8FEFF
    FFFFFFFF,F7E0C0C0808,
    F0E0C08,FFFFFFFFFFFF2F
    0C
330 DATA FFFFFFFFFCFCF8F8
    ,F0F0F0F0C,FFFFFFFFFF
    F8,FF3F3F3F3F3F1F,030
    30303030303 03
340 DATA FFFFFFFCF8F0C,FF
    FF,,,,,,FFFFFFFFFF3F1
    F1F,0F070301,FF7FFFFF
    FF3F1F0F,FFFEFE,,,,,F
    FFFFFFFFFFFFFFF
350 DATA DFDF9F9F9F9F9F9F
    ,9F9F9F9F9F8FCFC7,,,,
    ,,FFFFFFFFFFFFFFFF,00
    00000000030F3F,000000
    01071FFFFF
360 DATA 0000009 8FCFFFFFF
    ,F8FCFEFFFFFFFFFF,FFF
    FFFFFEFEC7E,C0E0E0F0
    C00080C,000000 80C0E0F
    0F
370 DATA FFFFFE7E1E0E,FCF
    CFCFCFCFCFCFC,0103070
    F1F3F7FFF,0103071F3FF
    FFFFF,000000000000FFF
    F
380 DATA 0000000000 3F3F1F
    ,1F0F0F0F1F3FFF,010
    30307 0F0F0F0F
390 CALL COLDR(1,7,1)
400 CALL COLOR(9,7,12)
410 CALL CDLOR(10,12,1)
420 CALL COLDR(11,12,3)
430 CALL COLOR(12,14,12)
440 CALL COLOR(13,14,3)
450 CALL COLOR(14,14,1)
460 CALL COLOR(15,3,1)
470 CALL COLOR(16,3,1)
480 RESTORE 490
490 DATA NEW YORK,ALBANY,
    7,25,PENNSYLVANIA,HAR
    RISBURG,10,21,NEW JER
    SEY,TRENTON,11,26
500 DATA WEST VIRGINIA,CH
    ARLESTON,15,17,MARYLA
    ND,ANNAPOLIS,13,23
510 DATA DELAWARE,DOVER,1
    4,25,VIRGINIA,RICHMON
    D,17,21
520 FDR L=1 TD 7
530 READ S$(L),CAP$(L),X(
    L),Y(L)
540 NEXT L
550 PRINT :"PRESS <ENTER>
     TO START.";
560 CALL KEY(0,K,S)
570 IF K<>13 THEN 560
580 CALL CLEAR
590 CALL SCREEN(8)
600 CALL COLOR(2,16,1)
610 PRINT TAB(16);"!#####
    ##"
620 PRINT TAB(16);"!#####
    #'p"
630 PRINT TAB(16);"!#####
    #adl"
```

```
640 PRINT TAB(16);"c#####bdm"
650 PRINT TAB(15);"hdxyz{3 SPACES}+n"
660 PRINT TAB(14);"iddddq{3 SPACES}-o"
670 PRINT TAB(13);"jdddsr{4 SPACES}/"
680 PRINT TAB(13);"kddt"
690 PRINT TAB(14);"wvu":::::::
700 CALL VCHAR(3,27,153,3)
710 CALL VCHAR(6,27,144,4)
720 CALL VCHAR(3,26,144,7)
730 CALL VCHAR(4,25,144,5)
740 CALL VCHAR(5,24,144,4)
750 CALL HCHAR(7,21,144,3)
760 CALL HCHAR(8,20,144,4)
770 RESTORE 820
780 FOR I=1 TO 28
790 READ R,C,G
800 CALL HCHAR(R,C,G)
810 NEXT I
820 DATA 3,25,154,4,24,154,6,23,155,6,22,156,6,21,156,6,20,157,7,20,158,8,19,159,10,27,152,10,28,146
830 DATA 10,29,147,13,22,128,13,23,136,13,24,136,14,22,129,14,23,136,14,24,137,15,23,130,15,24,138
840 DATA 16,25,131,16,24,151,16,23,144,17,24,150,17,23,149,18,23,148,18,24,147,18,14,146,17,15,145
850 CALL VCHAR(15,22,144,4)
860 CALL VCHAR(15,21,144,4)
870 CALL VCHAR(16,20,144,3)
880 CALL VCHAR(16,19,144,3)
890 CALL HCHAR(18,15,144,4)
900 FOR C=1 TO 7
910 T=0
920 RANDOMIZE
930 R=INT(7*RND)+1
940 IF S$(R)="" THEN 930
950 CALL GCHAR(X(R),Y(R),G)
960 CALL HCHAR(20,1,100,160)
970 FOR L=1 TO 7
980 CALL HCHAR(21,2+L,ASC(SEG$("STATE ?",L,1)))
990 CALL HCHAR(X(R),Y(R),32)
1000 CALL HCHAR(X(R),Y(R),63)
1010 NEXT L
1020 CALL HCHAR(21,11,100,15)
1030 S1$=""
1040 CALL SOUND(150,1397,2)
1050 FOR L=1 TO 15
1060 CALL KEY(0,K,S)
1070 IF S<1 THEN 1060
1080 IF K=13 THEN 1130
1090 IF K=6 THEN 1020
1100 CALL HCHAR(21,10+L,K)
1110 S1$=S1$&CHR$(K)
1120 NEXT L
1130 CALL SOUND(100,880,2)
1140 IF S$(R)=S1$ THEN 1260
1150 CALL SOUND(100,330,2)
1160 CALL SOUND(100,262,2)
1170 T=T+1
1180 IF T<2 THEN 1020
1190 CALL HCHAR(21,11,100,15)
1200 FOR L=1 TO LEN(S$(R))
1210 CALL HCHAR(21,10+L,ASC(SEG$(S$(R),L,1)))
1220 NEXT L
1230 GOSUB 1600
1240 C=C-1
1250 GOTO 1570
1260 GOSUB 1670
1270 FOR L=1 TO 9
1280 CALL HCHAR(23,2+L,ASC(SEG$("CAPITAL ?",L,1)))
1290 NEXT L
1300 T=0
1310 CALL HCHAR(23,13,100,15)
1320 S1$=""
1330 CALL SOUND(150,1397,2)
1340 FOR L=1 TO 15
1350 CALL KEY(0,K,S)
1360 IF S<1 THEN 1350
1370 IF K=13 THEN 1430
1380 IF K=6 THEN 1310
1390 CALL HCHAR(23,12+L,K)
1400 S1$=S1$&CHR$(K)
1410 NEXT L
1420 CALL SOUND(100,880,2)
1430 IF CAP$(R)=S1$ THEN 1550
1440 CALL SOUND(100,330,2)
1450 CALL SOUND(100,262,2)
1460 T=T+1
1470 IF T<2 THEN 1310
1480 CALL HCHAR(23,12,100,15)
1490 FOR L=1 TO LEN(CAP$(R))
1500 CALL HCHAR(23,12+L,ASC(SEG$(CAP$(R),L,1)))
1510 NEXT L
1520 GOSUB 1600
1530 C=C-1
1540 GOTO 1570
1550 GOSUB 1670
1560 S$(R)=""
1570 CALL HCHAR(X(R),Y(R),G)
1580 NEXT C
1590 GOTO 1720
1600 FOR L=1 TO 11
1610 CALL HCHAR(24,20+L,ASC(SEG$("PRESS ENTER",L,1)))
1620 NEXT L
1630 CALL KEY(0,K,S)
1640 IF K<>13 THEN 1630
1650 CALL HCHAR(24,21,100,11)
1660 RETURN
1670 CALL SOUND(100,262,2)
1680 CALL SOUND(100,330,2)
1690 CALL SOUND(100,392,2)
1700 CALL SOUND(200,523,2)
1710 RETURN
1720 CALL CLEAR
1730 PRINT "TRY AGAIN?  Y OR N"
1740 CALL KEY(0,K,S)
1750 IF K=89 THEN 480
1760 IF K<>78 THEN 1740
1770 CALL CLEAR
1780 END
```

©

COMPUTE!'s Guide To Typing In Programs

Before typing in any program, you should familiarize yourself with your computer. Learn how to use the keyboard to type in and correct BASIC programs. Read your manuals to understand how to save and load BASIC programs to and from your disk drive or cassette unit. Computers are precise—take special care to type the program *exactly* as listed, including any necessary punctuation and symbols, except for special characters as noted below. To help you with this task, we have implemented a special listing convention as well as a program to help check your typing—the "Automatic Proofreader." Please read the following notes before typing in any programs from COMPUTE!. They can save you a lot of time and trouble.

Commodore, Apple, and Atari programs can contain some hard-to-read (and hard-to-type) special characters, so we have developed a listing system that indicates the function of these control characters. (There are no special control characters in our IBM or TI-99/4A listings.) You will find Commodore and Atari special characters within curly braces; *do not type the braces*. For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. For Commodore, Apple, and Atari, a symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CTRL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple. Commodore computers also have a special control key labeled with the Commodore logo. Graphics characters entered with the Commodore logo key are enclosed in a special bracket that looks like this: [<A>]. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as $\underline{S}$. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6 $\underline{S}$}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (printed in white on black) should be entered after pressing the inverse video key.

Since spacing is sometimes important, any more than two spaces will be listed. For example, {6 SPACES} means to press the space bar six times. Our listings never leave a space at the end of a line, instead moving it to the next printed line as {SPACE}. For your convenience, we have prepared this quick-reference chart for the Commodore and Atari special characters:

Atari 400/800/XL/XE

When you see	Type	See	
{CLEAR}	ESC SHIFT <		Clear Screen
{UP}	ESC CTRL -		Cursor Up
{DOWN}	ESC CTRL =		Cursor Down
{LEFT}	ESC CTRL +		Cursor Left
{RIGHT}	ESC CTRL *		Cursor Right
{BACK S}	ESC DELETE		Backspace
{DELETE}	ESC CTRL DELETE		Delete character
{INSERT}	ESC CTRL INSERT		Insert character
{DEL LINE}	ESC SHIFT DELETE		Delete line
{INS LINE}	ESC SHIFT INSERT		Insert line
{TAB}	ESC TAB		TAB key
{CLR TAB}	ESC CTRL TAB		Clear tab
{SET TAB}	ESC SHIFT TAB		Set tab stop
{BELL}	ESC CTRL 2		Ring buzzer
{ESC}	ESC ESC		ESCape key

Commodore PET/CBM/VIC/64/128/16/+4

When You Read:	Press:	See:	When You Read:	Press:	See:
{CLR}	SHIFT CLR/HOME		[<1>]	COMMODORE 1	
{HOME}	CLR/HOME		[<2>]	COMMODORE 2	
{UP}	SHIFT ↑ CRSR ↓		[<3>]	COMMODORE 3	
{DOWN}	↑ CRSR ↓		[<4>]	COMMODORE 4	
{LEFT}	SHIFT ← CRSR →		[<5>]	COMMODORE 5	
{RIGHT}	← CRSR →		[<6>]	COMMODORE 6	
{RVS}	CTRL 9		[<7>]	COMMODORE 7	
{OFF}	CTRL 0		[<8>]	COMMODORE 8	
{BLK}	CTRL 1		{ F1 }	f1	
{WHT}	CTRL 2		{ F2 }	SHIFT f1	
{RED}	CTRL 3		{ F3 }	f3	
{CYN}	CTRL 4		{ F4 }	SHIFT f3	
{PUR}	CTRL 5		{ F5 }	f5	
{GRN}	CTRL 6		{ F6 }	SHIFT f5	
{BLU}	CTRL 7		{ F7 }	f7	
{YEL}	CTRL 8		{ F8 }	SHIFT f7	
			←	←	

The Automatic Proofreader

We have developed a series of simple, yet effective programs that can help check your typing. Type in the appropriate Proofreader program listed below, then save it for future use. On the VIC, 64, or Atari, run the Proofreader to activate it, then enter NEW to erase the BASIC loader (the Proofreader remains active, hidden in memory, as a machine language program). Pressing RUN/STOP–RESTORE or SYSTEM RESET deactivates the Proofreader. You can use SYS 886 to reactivate the VIC/64 Proofreader, or PRINT USR(1536) to reenable the Atari Proofreader. On the Apple, the Proofreader automatically erases the BASIC portion of itself after you activate it by typing RUN, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program. The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a decimal number (on the Commodore), a hexadecimal number (on the Apple), or a pair of letters (on the Atari or IBM) appears. The number or pair of letters is called a *checksum*. Try making a change in the line, and notice how the checksum changes.

All you need to do is compare the value provided by the Proofreader with the checksum printed in the program listing in the magazine. In Commodore listings, the checksum is a number from 0 to 255. It is set off from the rest of the line with *rem*. This prevents a syntax error if the checksum is typed in, but the REM statements and checksums need *not* be typed in. It is just there for your information.

In Atari, Apple, and IBM listings, the checksum is given to the left of each line number. Just type in the program one line at a time (without the printed checksum) and compare the checksum generated by the Proofreader to the checksum in the listing. If they match, go on to the next line. If not, check your typing: You've made a mistake. On the Commodore, Atari, and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Commodore and Atari Proofreaders do not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. The IBM Proofreader is the pickiest of all; it *will* detect errors in spacing and transposition. Be sure to leave Caps Lock on, except when typing lowercase characters.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you type NEW, the Proofreader prompts you to press Y to be sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program in BASIC as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to resave it to disk. The version of your program that you resave from BASIC will take up less space on disk and will load faster, but it can no longer be edited with the Proofreader. If you want to convert a program to Proofreader format, save it to disk with SAVE *"filename"*,A.

Special Proofreader Notes For Commodore Cassette Users

The Proofreader resides in a section of memory called the cassette buffer, which is used during tape LOADs and SAVEs. Therefore, be sure to press RUN/STOP–RESTORE to get the Proofreader out of the way before saving or loading a program. If you want to use the Proofreader with tape, run the Proofreader, then enter these two lines *exactly* as shown, pressing RETURN after each one:

```
A$="PROOFREADER.T":B$="{10
 SPACES}":FOR X=1 TO 4:A$=A$
 +B$:NEXT
FOR X=886 TO 1018:A$=A$+CHR$
(PEEK(X)):NEXT:OPEN 1,1,1,A$:
CLOSE1
```

Then insert a blank tape and press RECORD and PLAY to save a special version of the Proofreader. Anytime you need to reload the Proofreader after it has been erased—for example, after you reload a paritally completed program—just rewind the tape, type OPEN1:CLOSE1, then press PLAY. You'll see the message FOUND PROOFREADER.T, but not the familiar LOADING message. Don't worry; the Proofreader is in memory. When READY comes back, enter SYS 886.

Program 1: VIC/64 Proofreader

By Charles Brannon, Program Editor

```
10 PRINT"{CLR}PLEASE WAIT...":
   FORI=886TO1018:READA:CK=CK+
   A:POKEI,A:NEXT
20 IF CK<>17539 THEN PRINT"
   {DOWN}YOU MADE AN ERROR":PR
   INT"IN DATA STATEMENTS.":EN
   D
30 SYS886:PRINT"{CLR}{2 DOWN}P
   ROOFREADER ACTIVATED.":NEW
40 DATA 173,036,003,201,150,20
   8,001,096,141,151,003,173
50 DATA 037,003,141,152,003,16
   9,150,141,036,003,169,003
60 DATA 141,037,003,169,000,13
   3,254,096,032,087,241,133
70 DATA 251,134,252,132,253,00
   8,201,013,240,017,201,032
80 DATA 240,005,024,101,254,13
   3,254,165,251,166,252,164
90 DATA 253,040,096,169,013,03
   2,210,255,165,214,141,251
100 DATA 003,206,251,003,169,0
    00,133,216,169,019,032,210
110 DATA 255,169,018,032,210,2
    55,169,58,032,210,255,166
120 DATA 254,169,000,133,254,1
    72,151,003,192,087,208,006
130 DATA 032,205,189,076,235,0
    03,032,205,221,169,032,032
140 DATA 210,255,032,210,255,1
    73,251,003,133,214,076,173
150 DATA 003
```

Program 2: Atari Proofreader

By Charles Brannon, Program Editor

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:RE
    AD A:POKE I,A:CK=CK+A
    :NEXT I
120 IF CK<>19072 THEN ? "
    Error in DATA Stateme
    nts.  Check Typing.":
    END
130 A=USR(1536)
140 ? :? "Automatic Proof
    reader Now Activated.
    "
150 END
160 DATA 104,160,0,185,26
    ,3,201,69,240,7
170 DATA 200,200,192,34,2
    08,243,96,200,169,74
180 DATA 153,26,3,200,169
    ,6,153,26,3,162
190 DATA 0,189,0,228,157,
    74,6,232,224,16
200 DATA 208,245,169,93,1
    41,78,6,169,6,141
210 DATA 79,6,24,173,4,22
    8,105,1,141,95
```

```
220 DATA 6,173,5,228,105,
    0,141,96,6,169
230 DATA 0,133,203,96,247
    ,238,125,241,93,6
240 DATA 244,241,115,241,
    124,241,76,205,238
250 DATA 0,0,0,0,0,32,62,
    246,8,201
260 DATA 155,240,13,201,3
    2,240,7,72,24,101
270 DATA 203,133,203,104,
    40,96,72,152,72,138
280 DATA 72,160,0,169,128
    ,145,88,200,192,40
290 DATA 208,249,165,203,
    74,74,74,74,24,105
300 DATA 161,160,3,145,88
    ,165,203,41,15,24
310 DATA 105,161,200,145,
    88,169,0,133,203,104
320 DATA 170,104,168,104,
    40,96
```

Program 3: IBM Proofreader

By Charles Brannon, Program Editor

```
10 'Automatic Proofreader Ver
   sion 2.00 (Lines 270,510,5
   15,517,620,630 changed fro
   m V1.0)
100 DIM L$(500),LNUM(500):COL
    OR 0,7,7:KEY OFF:CLS:MAX=
    0:LNUM(0)=65536!
110 ON ERROR GDTO 120:KEY 15,
    CHR$(4)+CHR$(70):ON KEY(1
    5) GDSUB 640:KEY (15) ON:
    GDTO 130
120 RESUME 130
130 DEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:P
    RINT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN-IN
    T(LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:PD
    KE 1052,34:POKE 1054,0:PO
    KE 1055,79:PDKE 1056,13:P
    OKE 1057,28:LINE INPUT L$
    :OEF SEG:IF L$="" THEN 15
    0
170 IF LEFT$(L$,1)=" " THEN L
    $=MID$(L$,2):GOTD 170
180 IF VAL(LEFT$(L$,2))=0 AND
     MID$(L$,3,1)=" " THEN L$
    =MIO$(L$,4)
190 LNUM=VAL(L$):TEXT$=MIO$(L
    $,LEN(STR$(LNUM))+1)
200 IF ASC(L$)>57 THEN 260 'n
    o line number, therefore
    command
210 IF TEXT$="" THEN GOSUB 54
    0:IF LNUM=LNUM(P) THEN GO
    SUB 560:GOTO 150 ELSE 150
220 CKSUM=0:FOR I=1 TD LEN(L$
    ):CKSUM=(CKSUM+ASC(MIO$(L
    $,I))*I) AND 255:NEXT:LOC
    ATE Y,1:PRINT CHR$(65+CKS
    UM/16)+CHR$(65+(CKSUM AND
     15))+" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM
     THEN L$(P)=TEXT$:GOTO 15
    0 'replace line
240 GOSUB 580:GOTO 150 'inser
    t the line
260 TEXT$="":FOR I=1 TO LEN(L
    $):A=ASC(MID$(L$,I)):TEXT
    $=TEXT$+CHR$(A+32*(A>96 A
    ND A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," "
    ):COMMAND$=TEXT$:ARG$="":
    IF DELIMITER THEN COMMAND
    $=LEFT$(TEXT$,DELIMITER-1
    ):ARG$=MID$(TEXT$,DELIMIT
    ER+1) ELSE OELIMITER=INST
    R(TEXT$,CHR$(34)):IF DELI
    MITER THEN COMMAND$=LEFT$
    (TEXT$,DELIMITER-1):ARG$=
    MID$(TEXT$,DELIMITER)
280 IF COMMAND$<>"LIST" THEN
    410
290 OPEN "scrn:" FOR OUTPUT A
    S #1
300 IF ARG$="" THEN FIRST=0:P
    =MAX-1:GOTO 340
310 DELIMITER=INSTR(ARG$,"-")
    :IF DELIMITER=0 THEN LNUM
    =VAL(ARG$):GOSUB 540:FIRS
    T=P:GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELI
    MITER)):LAST=VAL(MID$(ARG
    $,DELIMITER+1))
330 LNUM=FIRST:GOSUB 540:FIRS
    T=P:LNUM=LAST:GOSUB 540:I
    F P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(
    STR$(LNUM(X)),2)+" "
350 IF CKFLAG=0 THEN A$="":GO
    TD 370
360 CKSUM=0:A$=N$+L$(X):FOR I
    =1 TO LEN(A$):CKSUM=(CKSU
    M+ASC(MID$(A$,I))*I) AND
    255:NEXT:A$=CHR$(65+CKSUM
    /16)+CHR$(65+(CKSUM AND 1
    5))+" "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LLIST" THEN
    OPEN "lpt1:" FOR OUTPUT A
    S #1:GOTO 300
420 IF COMMAND$="CHECK" THEN
    CKFLAG=1:GOTO 290
430 IF COMMAND$<>"SAVE" THEN
    450
440 GOSUB 600:OPEN ARG$ FDR O
    UTPUT AS #1:ARG$="":GDTO
    300
450 IF CDMMAND$<>"LOAD" THEN
    490
460 GOSUB 600:OPEN ARG$ FOR I
    NPUT AS #1:MAX=0:P=0
470 WHILE NOT EDF(1):LINE INP
    UT #1,L$:LNUM(P)=VAL(L$):
    L$(P)=MID$(L$,LEN(STR$(VA
    L(L$)))+1):P=P+1:WEND
480 MAX=P:CLDSE #1:GDTD 130
490 IF COMMAND$="NEW" THEN IN
    PUT "Erase program - Are
    you sure";L$:IF LEFT$(L$,
    1)="y" OR LEFT$(L$,1)="Y"
     THEN MAX=0:GOTO 130:ELSE
     130
500 IF COMMAND$="BASIC" THEN
    COLOR 7,0,0:ON ERROR GOTD
     0:CLS:END
510 IF CDMMAND$<>"FILES" THEN
     520
515 IF ARG$="" THEN ARG$="A:"
     ELSE SEL=1:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTD
    130
540 P=0:WHILE LNUM>LNUM(P) AN
    D P<MAX:P=P+1:WEND:RETURN
560 MAX=MAX-1:FOR X=P TO MAX:
    LNUM(X)=LNUM(X+1):L$(X)=L
    $(X+1):NEXT:RETURN
580 MAX=MAX+1:FOR X=MAX TO P+
    1 STEP -1:LNUM(X)=LNUM(X-
    1):L$(X)=L$(X-1):NEXT:L$(
    P)=TEXT$:LNUM(P)=LNUM:RET
    URN
600 IF LEFT$(ARG$,1)<>CHR$(34
    ) THEN 520 ELSE ARG$=MID$
    (ARG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34
    ) THEN ARG$=LEFT$(ARG$,LE
    N(ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$,"
    .")=0 THEN ARG$=ARG$+".BA
    S"
630 SEL=0:RETURN
640 CLDSE #1:CKFLAG=0:PRINT"S
    topped.":RETURN 150
650 PRINT "Error #";ERR:RESUM
    E 150
```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```
10 C = 0: FOR I = 768 TO 768 +
   68: READ A:C = C + A: POKE I
   ,A: NEXT
20 IF C < > 7258 THEN PRINT "ER
   ROR IN PROOFREADER DATA STAT
   EMENTS": END
30 IF PEEK (190 * 256) < > 76 T
   HEN POKE 56,0: POKE 57,3: CA
   LL 1002: GOTO 50
40 PRINT CHR$ (4);"IN#A$300"
50 POKE 34,0: HOME : PDKE 34,1:
    VTAB 2: PRINT "PROOFREADER
   INSTALLEO"
60 NEW
100 DATA 216,32,27,253,201,141
110 DATA 208,60,138,72,169,0
120 DATA 72,189,255,1,201,160
130 DATA 240,8,104,10,125,255
140 DATA 1,105,0,72,202,208
150 DATA 238,104,170,41,15,9
160 DATA 48,201,58,144,2,233
170 DATA 57,141,1,4,138,74
180 DATA 74,74,74,41,15,9
190 DATA 48,201,58,144,2,233
200 DATA 57,141,0,4,104,170
210 DATA 169,141,96
```

©

MLX Machine Language Entry Program For Commodore 64

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. *You need to know nothing about machine language to use MLX—it was designed for everyone. At least 8K expansion memory is required.*

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file.

Using MLX

Type in and save the appropriate version of MLX (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX for the 64 asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program.

When you run MLX, you'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number*. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the INST/DEL key; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the space bar or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

	U	I	O			7	8	9
H	J	K	L	become	0	4	5	6
	M	,	.			1	2	3

64 MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. MLX recognizes these commands:

SHIFT-S: Save
SHIFT-L: Load
SHIFT-N: New Address
SHIFT-D: Display

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk, as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember what address you stop at. The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press SHIFT-L to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

64 MLX: Machine Language Entry

```
10 REM LINES CHANGED FROM MLX
   {SPACE}VERSION 2.00 ARE 750
   ,765,770 AND 860       :rem 50
20 REM LINE CHANGED FROM MLX V
   ERSION 2.01 IS 300 :rem 147
100 PRINT"{CLR}E63";CHR$(142);
    CHR$(8);:POKE53281,1:POKE5
    3280,1                :rem 67
101 POKE 788,52:REM DISABLE RU
    N/STOP               :rem 119
110 PRINT"{RVS}{39 SPACES}";
                         :rem 176
120 PRINT"{RVS}{14 SPACES}
    {RIGHT}{OFF}E*3£{RVS}
    {RIGHT} {RIGHT}{2 SPACES}
    E*3{OFF}E*3£{RVS}£{RVS}
    {14 SPACES}";        :rem 250
130 PRINT"{RVS}{14 SPACES}
    {RIGHT} EG3{RIGHT}
    {2 RIGHT} {OFF}£{RVS}£
    E*3{OFF}E*3{RVS}
    {14 SPACES}";         :rem 35
140 PRINT"{RVS}{41 SPACES}"
                         :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK} M
    ACHINE LANGUAGE EDITOR VER
    SION 2.02{5 DOWN}":rem 238
210 PRINT"E53{2 UP}STARTING AD
    DRESS?{8 SPACES}{9 LEFT}";
                         :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+11
    9*F)                 :rem 166
220 IFS<256OR(S>40960ANDS<4915
    2)ORS>53247THENGOSUB3000:G
    OTO210               :rem 235
225 PRINT:PRINT:PRINT :rem 180
230 PRINT"E53{2 UP}ENDING ADDR
    ESS?{8 SPACES}{9 LEFT}";:I
    NPUTE:F=1-F:C$=CHR$(31+119
    *F)                   :rem 20
240 IFE<256OR(E>40960ANDE<4915
    2)ORE>53247THENGOSUB3000:G
    OTO230               :rem 183
250 IFE<STHENPRINTC$;"{RVS}END
    ING < START{2 SPACES}":GOS
    UB1000:GOTO 230     :rem 176
260 PRINT:PRINT:PRINT :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S
                          :rem 56
310 A=1:PRINTRIGHT$("0000"+MID
    $(STR$(AD),2),5);":";
                          :rem 33
315 FORJ=ATO6             :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:G
    OTO320               :rem 228
390 IFN=-211THEN 710      :rem 62
400 IFN=-204THEN 790      :rem 64
410 IFN=-206THENPRINT:INPUT"
    {DOWN}ENTER NEW ADDRESS";Z
    Z                     :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETH
    ENPRINT"{RVS}OUT OF RANGE"
    :GOSUB1000:GOTO410:rem 225
417 IFN=-206THENAD=ZZ:PRINT:GO
    TO310                :rem 238
420 IF N<>-196 THEN 480
                         :rem 133
430 PRINT:INPUT"DISPLAY:FROM";
    F:PRINT,"TO";:INPUTT
                         :rem 234
440 IFF<SORF>EORT<SORT>ETHENPR
    INT"AT LEAST";S;"{LEFT}, N
    OT MORE THAN";E:GOTO430
                         :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINT
    RIGHT$("0000"+MID$(STR$(I)
    ,2),5);":";           :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRIN
    TRIGHT$("00"+MID$(STR$(N),
    2),3);",";            :rem 66
```

```
460 GETA$:IFA$>""THENPRINT:PRI
    NT:GOTO310            :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI
    :PRINT:PRINT:GOTO310
                          :rem 50
480 IFN<0 THEN PRINT:GOTO310
                         :rem 168
490 A(J)=N:NEXTJ         :rem 199
500 CKSUM=AD-INT(AD/256)*256:F
    ORI=1TO6:CKSUM=(CKSUM+A(I)
    )AND255:NEXT         :rem 200
510 PRINTCHR$(18);:GOSUB570:PR
    INTCHR$(146);         :rem 94
511 IFN=-1THENA=6:GOTO315
                         :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHE
    N530                 :rem 122
520 PRINT:PRINT"LINE ENTERED W
    RONG : RE-ENTER":PRINT:GOS
    UB1000:GOTO310       :rem 176
530 GOSUB2000            :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):
    NEXT:POKE54272,0:POKE54273
    ,0                   :rem 227
550 AD=AD+6:IF AD<E THEN 310
                         :rem 212
560 GOTO 710             :rem 108
570 N=0:Z=0               :rem 88
580 PRINT"[£]";           :rem 81
581 GETA$:IFA$=""THEN581
                          :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*
    (A$=".")-4*(A$="J")-5*(A$=
    "K")-6*(A$="L")       :rem 41
583 AV=AV-7*(A$="U")-8*(A$="I"
    )-9*(A$="O"):IFA$="H"THENA
    $="0"                :rem 134
584 IFAV>0THENA$=CHR$(48+AV)
                         :rem 134
585 PRINTCHR$(20);:A=ASC(A$):I
    FA=13ORA=44ORA=32THEN670
                         :rem 229
590 IFA>128THENN=-A:RETURN
                         :rem 137
600 IFA<>20 THEN 630      :rem 10
610 GOSUB690:IFI=1ANDT=44THENN
    =-1:PRINT"{OFF}{LEFT}
    {LEFT}";:GOTO690      :rem 62
620 GOTO570              :rem 109
630 IFA<48ORA>57THEN580
                         :rem 105
640 PRINTA$;:N=N*10+A-48
                         :rem 106
650 IFN>255 THEN A=20:GOSUB100
    0:GOTO600            :rem 229
660 Z=Z+1:IFZ<3THEN580   :rem 71
670 IFZ=0THENGOSUB1000:GOTO570
                         :rem 114
680 PRINT",";:RETURN     :rem 240
690 S%=PEEK(209)+256*PEEK(210)
    +PEEK(211)           :rem 149
691 FORI=1TO3:T=PEEK(S%-I)
                          :rem 67
695 IFT<>44ANDT<>58THENPOKES%-
    I,32:NEXT            :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1)
    ;:RETURN               :rem 7
710 PRINT"{CLR}{RVS}*** SAVE *
    **{3 DOWN}"          :rem 236
715 PRINT"{2 DOWN}(PRESS {RVS}
    RETURN{OFF} ALONE TO CANCE
    L SAVE){DOWN}"       :rem 106
720 F$="":INPUT"{DOWN} FILENAM
    E";F$:IFF$=""THENPRINT:PRI
    NT:GOTO310            :rem 71
730 PRINT:PRINT"{2 DOWN}{RVS}T
    {OFF}APE OR {RVS}D{OFF}ISK
    : (T/D)"             :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"T
    HEN740                :rem 36
750 DV=1-7*(A$="D"):IFDV=8THEN
    F$="0:"+F$:OPEN15,8,15,"S"
    +F$:CLOSE15          :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK
    (54)-LEN(T$):POKE782,ZK/25
    6                      :rem 3
762 POKE781,ZK-PEEK(782)*256:P
    OKE780,LEN(T$):SYS65469
                         :rem 109
763 POKE780,1:POKE781,DV:POKE7
    82,1:SYS65466         :rem 69
765 K=S:POKE254,K/256:POKE253,
    K-PEEK(254)*256:POKE780,25
    3                     :rem 17
766 K=E+1:POKE782,K/256:POKE78
    1,K-PEEK(782)*256:SYS65496
                         :rem 235
770 IF(PEEK(783)AND1)OR(191AND
    ST)THEN780           :rem 111
775 PRINT"{DOWN}DONE.{DOWN}":G
    OTO310               :rem 113
780 PRINT"{DOWN}ERROR ON SAVE.
    {2 SPACES}TRY AGAIN.":IFDV
    =1THEN720            :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E
    2$:PRINTE1$;E2$:CLOSE15:GO
    TO720                :rem 103
790 PRINT"{CLR}{RVS}*** LOAD *
    **{2 DOWN}"          :rem 212
795 PRINT"{2 DOWN}(PRESS {RVS}
    RETURN{OFF} ALONE TO CANCE
    L LOAD)"              :rem 82
800 F$="":INPUT"{2 DOWN} FILEN
    AME";F$:IFF$=""THENPRINT:G
    OTO310               :rem 144
810 PRINT:PRINT"{2 DOWN}{RVS}T
    {OFF}APE OR {RVS}D{OFF}ISK
    : (T/D)"             :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"T
    HEN820                :rem 34
830 DV=1-7*(A$="D"):IFDV=8THEN
    F$="0:"+F$           :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK
    (54)-LEN(T$):POKE782,ZK/25
    6                      :rem 2
841 POKE781,ZK-PEEK(782)*256:P
    OKE780,LEN(T$):SYS65469
                         :rem 107
845 POKE780,1:POKE781,DV:POKE7
    82,1:SYS65466         :rem 70
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(191AND
    ST)THEN870           :rem 111
865 PRINT"{DOWN}DONE.":GOTO310
                          :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.
    {2 SPACES}TRY AGAIN.{DOWN}
    ":IFDV=1THEN800      :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E
    2$:PRINTE1$;E2$:CLOSE15:GO
    TO800                :rem 102
1000 REM BUZZER          :rem 135
1001 POKE54296,15:POKE54277,45
     :POKE54278,165      :rem 207
1002 POKE54276,33:POKE 54273,6
     :POKE54272,5         :rem 42
1003 FORT=1TO200:NEXT:POKE5427
     6,32:POKE54273,0:POKE5427
     2,0:RETURN          :rem 202
2000 REM BELL SOUND       :rem 78
2001 POKE54296,15:POKE54277,0:
     POKE54278,247       :rem 152
2002 POKE 54276,17:POKE54273,4
     0:POKE54272,0        :rem 86
2003 FORT=1TO100:NEXT:POKE5427
     6,16:RETURN          :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PA
     GE OR ROM":GOTO1000
                          :rem 89
```

©

CAPUTE!

Modifications or Corrections To Previous Articles

Commodore 64 Disk Commander

This program from the September issue (p. 80) has a bug in the DOPEN command. *Do not* use DOPEN until you correct the problem. If you disassemble the relocated "Disk Commander" code, you'll notice the instruction CMP $062D,X at location $A797 in the DOPEN routine. The instruction should be CMP $026D,X. This portion of the routine is intended to assign a unique secondary address to each opened file, but the bug causes all files to be opened with the same secondary address. If you never DOPEN more than one file at a time, there's no problem. However, multiple DOPENs lead to improperly closed files, which are denoted in the disk directory by an asterisk next to the filename. If you see any of these so-called poison files on your disk, you should remove them with the VALIDATE command (OPEN 1,8,15,"V0:" :CLOSE 1), not the SCRATCH command.

To fix "Disk Commander," first enter POKE 44,20:POKE 5120,0:NEW to reconfigure memory, then load and run "MLX". Use the MLX Load command (SHIFT-L) to load your existing version of Disk Commander. Next, use the New Address command (SHIFT-N) to move to line 3591, then enter the following new data:

```
3591 :048,007,221,109,002,240,122
```

Now press SHIFT-S to call the MLX Save feature and save a copy of the corrected program.

Our thanks to reader Franz Paulsen for uncovering this bug.

Atari Animation With P/M Graphics, Part 2

Part of line 90 is missing in the program example in the first column on page 102 of this article from the October issue. It should read as follows:

```
NS 90 FOR X=PMBASE+1024 TO PMBASE+2048:
     POKE X,0:NEXT X
```

The Last Warrior

A number of readers have had difficulties with line 480 in the IBM version of this game program from the September issue (p. 54). The first character within quotes in that line is the lowercase letter l, *not* the numeral 1. The two characters do have a similar appearance in the listing, but 164 is not a reasonable parameter for a PLAY statement, while l64 is. ©

Word Processor For IBM

Professional Software has introduced *Write 'n Spell*, a $149 word processor which contains an integrated 90,000-word dictionary that checks and corrects spelling. In addition to standard features found in most sophisticated word processors, *Write 'n Spell* also contains built-in mail-merge, graphing, and simultaneous typing-while-printing capabilities. The program will merge with *Lotus 1-2-3*, *pfs:File*, *Multiplan*, and many other popular application programs. The word processor also has pull-down HELP and OPTION windows.

Write 'n Spell is available for the IBM PC, PCjr, AT, and compatible computers.

Professional Software, Inc., 51 Fremont Street, Needham, MA 02194.

Circle Reader Service Number 220.

World War II Air War

Strategic Studies Group (SSG), creators of *Reach for the Stars* and *Carriers at War* strategy games, has released *Europe Ablaze: The Air War Over England and Germany 1939–1945*. This historical simulation contains three major scenarios, selected from the various phases of the air war, and also a game design kit that lets you create your own scenarios. Major bombing missions are planned twice each day, and players are required to select targets, plot course and speed, determine H-hour, and allocate squadrons. Fighter aircraft patrol and intercept in response to ground and radar sightings.

Europe Ablaze is available for Apple II-series computers (with 64K RAM) and for the Commodore 64, at a suggested retail price of $50.

Strategic Studies Group, 1747 Orleans Court, Walnut Creek, CA 94598.

Circle Reader Service Number 221.

Apple II Spreadsheet

Mouse Calc, a mouse-controlled spreadsheet for the Apple IIc and 128K Apple IIe, has been announced by International Solutions, Inc. The program includes integrated graphics, mouse-operated editing and selecting techniques, pull-down menus, and color display. *Mouse Calc* is the first in a series of application programs from International Solutions.

Users can perform 24 of the most commonly used arithmetic, logical, search, and other spreadsheet functions with *Mouse Calc*. The program provides rounding and logical functions such as AND/OR and TRUE/FALSE. *Mouse Calc* can merge two or more files, and it can read files created with *VisiCalc*, *AppleWorks*, and other programs using the DIF format.

The program requires a mouse controller, such as the AppleMouse II, and a second disk drive is recommended. Suggested retail is $149.95, and includes a 90-day warranty.

International Solutions, Inc., 910 West Maude Avenue, Sunnyvale, CA 94086.

Circle Reader Service Number 222.

Educational, Entertainment Programs

Among the software titles recently introduced by CBS Software are several educational and entertainment programs. Included are *The Body in Focus*, a self-paced color-graphics human anatomy program for the Apple II+, IIe, IIc, Commodore 64, and IBM PC ($39.95 each); *Success with Math*, a series of math tutorials for ages 6 through 18 for the Apple II+, IIe, IIc, Commodore 64, and IBM PC/PCjr ($24.95 each); *Success with Algebra*, a similar series covering algebra for grades 7–12 for the Apple II series, Commodore 64, and IBM ($34.95); and *Quink*, a game of pattern recognition and knowledge for ages ten and older, for the Apple II series, 64, and IBM PC/PCjr ($34.95).

CBS Software, One Fawcett Place, Greenwich, CT 06836.

Circle Reader Service Number 223.

64 Bulletin Board

Bozart Co. has introduced two telecommunications packages for the Commodore 64: *Bozboard*, a full-featured bulletin board program, and *Bozterm*, an all-purpose terminal program. *Bozboard* is set to run with one or two 1541 disk drives or the MSD SD-2 dual drive. It is compatible with the Commodore 1650 Automodem and also with the Westridge 6420, TeleLearning, Mitey Mo, and HES II modems. The system requires a printer.

With *Bozboard* ($40), you have a choice of eight subboards, public messages, electronic mail, uploading, downloading, and a magazine feature which allows the system operator (sysop) to publish a color/graphics electronic magazine on the BBS. The program transfers files using the standard XMODEM protocol and its own Bozart protocol. The Bozart protocol is capable of transferring high-resolution graphics and allows the bulletin board user to view the graphics screen as it is downloaded.

Bozterm ($20) offers the user the option of automatic dialing or manual dialing. Seven of the function keys can be defined to automatically transmit any 80-character message, read the disk directory, upload buffer contents, capture incoming data to the buffer, print it to the screen or a printer, or save it to disk as an edited or unedited file.

Bozart Co., 7818 Summerfield Road, Summerfield, NC 27358.

Circle Reader Service Number 224.

Atari Interface

Integrated Computer Equipment Company (ICECO) has introduced the ICE-PIC (ICE's Parallel Interface Converter), a printer interface for Atari computers which also includes graphics software drivers.

The ICEPIC converts parallel-interface (Centronics-compatible) printers to a joystick interface (joystick port 2 or 4) which can be used by Atari 400/800 and XL computers with no hardware modifications. The ICEPIC requires no 850 Interface Module, no cable, and no external power supply. The software supports any printer in text mode and provides graphic functions for Epson or Okidata 92/93 graphics compatible printers. The ICE-PIC works with most Atari programs, such as most BASIC programs, *AtariWriter*, *Letter Perfect*, *B/GRAPH*, Koala *Micro Illustrator*, and *AtariArtist*. Several utility programs are included with

the software, including a diagnostic checkout program, a warm reboot program, and a *MicroPainter* file display program.

Suggested retail is $49.95 for the interface, software, and manual. There is a 30-day money back guarantee, a 90-day replacement warranty, and a lifetime $19 repair/replacement policy.

Integrated Computer Equipment Company, 8507 Natural Bridge Road, St. Louis, MO 63121.

Circle Reader Service Number 225.

64, Apple Karate

Data East has converted its arcade action game, *Karate Champ*, to a new computer version for the Commodore 64 and Apple II series. The $29.95 game features two-player and player-versus-computer modes as you guide your karate fighter through successive matches. Using the joystick, you can make your fighter lunge, kick, spin, somersault, reverse-punch, and block.

Data East USA, Inc., 470 Gianni Street, Santa Clara, CA 95054.

Circle Reader Service Number 226.

Pascal For 64 & 128

A complete Pascal development system for the Commodore 64 and 128 has been released by Abacus Software. *Super Pascal* includes an extensive compiler, a source file editor, an integrated assembler, and a comprehensive utility package for file and disk management.

Also included are a handbook with more than 200 pages and a systems disk. Suggested retail price is $59.95.

Abacus Software, 2201 Kalamazoo SE, P.O. Box 7211, Grand Rapids, MI 49510.

Circle Reader Service Number 227.

New From Epson

Epson has developed several new printers for home users. Among these are the DX-10 ($399), a daisywheel printer which prints at ten characters per second (cps); and the DX-20 ($499), a daisywheel with a 1K byte print buffer, 20 characters-per-second (cps) print speed, and a Diablo All Purpose Interface (RS-232C, IEEE-488, and parallel).

Also new from Epson is the Spectrum LX-90, a dot-matrix printer with draft and near-letter-quality (NLQ) modes. It comes with a printer interface cartridge that makes it ready for use with the IBM PC, PCjr, or Apple IIc. Draft copy is printed at 100 cps; NLQ at 16 cps. The Spectrum LX-90 retails for $389.

Comrex, a division of Epson, has released the CR-128 intelligent printer buffer. Features include 128K buffer memory and built-in serial-to-serial, serial-to-parallel, parallel-to-parallel, and parallel-to-serial interfaces. Suggested retail is $299 Another new Comrex product is the CR-700 series of bidirectional A-B switch boxes, which simplify connections to the computer and eliminate the need to swap cables when changing peripherals. The switch boxes retail for $39.95.

Epson America, 2780 Lomita Blvd., Torrance, CA 90505.

Circle Reader Service Number 228.

World War II Combat Game

Under Fire, from Avalon Hill, combines the depth of a war game simulation with the colorful graphics of an arcade game. Authentic armies, weapons, and maps from World War II add to the game's realism. Different maps and scenarios are included on disk; players can also create their own.

A sample screen from Avalon Hill's Under Fire *strategy game.*

Under Fire is available for the Apple II series. A joystick is optional for the Apple IIc and IIe, but required for the II+. Suggested retail price is $59.95.

Avalon Hill Game Company, 4517 Hartford Rd., Baltimore, MD 21214.

Circle Reader Service Number 229.

TI Disk Organizer

TI programs can be organized on a single disk with *Disk Data Base* from Asgard Software. The program lets you sort and print a catalog by either disk name or filename, to print it out unsorted, or to selectively print out all entries that contain a certain string. The catalog can also be broken up into blocks of 250 entries for easy management.

Data files can be converted from the Master Disk File to the *Disk Data Base* format. Also featured are numerous prompts and an online dictionary of terms. *Disk Data Base* requires Extended BASIC, a 32K memory expansion unit, and a disk drive and controller. A printer and second disk drive are recommended. Price: $15.

Asgard Software, P.O. Box 10306, Rockville, MD 20850.

Circle Reader Service Number 230.

Bulletin Board Directory

A new directory of computer bulletin boards, called *Plumbline*, is now available from the publishers of *Plumb*, a newsletter about personal telecommunications. The directory lists over 1,000 bulletin boards available to the public. Each entry includes a brief description of the bulletin board, the type of computer it runs on, and its primary area of interest.

Plumbline is included with a subscription to *Plumb*, $26.50; or can be purchased separately for $8.

Plumb, P.O. Box 300, Harrods Creek, KY 40027

Circle Reader Service Number 231.

Pascal Tutorial For Apple

Wiley Software's new *Visible Pascal* uses graphics, word processing, and music to teach the Pascal programming language on Apple II computers. Programs are displayed while they're being created, at a speed controlled by the programmer. The system has more than 80 error messages for pointing out mistakes. Users can create "productions," with animated characters and a soundtrack. The package also includes 56 sample programs that are ready to run.

No prior computing knowledge is needed. *Visible Pascal* runs on Apple II-series computers with at least 64K RAM. A joystick is recommended.

Wiley Professional Software, 605 Third Ave., New York, NY 10158.

Circle Reader Service Number 232.

Boolean Games

Sunburst has introduced *High Wire Logic*, a game for teaching Boolean logic to youngsters in grades 5 through 12. Two sets of colored shapes appear on the screen: one on a high wire and another set that falls to the net below. Using the logical functions AND, OR, AND-AND, OR-OR, and EXCLUSIVE OR, students earn points by writing rules to fit the shapes on the high wire but not the shapes in the net.

High Wire Logic is available for Apple II computers with at least 48K RAM; retail price is $59.

Sunburst Communications, Inc., 39 Washington Ave., Pleasantville, NY 10570.

Circle Reader Service Number 233.

Life/Time Manager

A new program from Psychometric

Software provides assistance in identifying goals and organizing time. Developed by a psychiatrist, *Life/Time Manager* is based on psychological and time management principles. It consists of three sections: Goals, Activities, and Schedules. Included are a prioritized daily To Do List and a weekly schedule analysis.

The program runs on the IBM PC, PCjr, or AT, with at least 128K RAM; or on the Apple II+, IIc, or IIe. Suggested price is $49.95.

Psychometric Software, Inc., 2050 S. Patrick Dr., Indian Harbour Beach, FL 32937.

Circle Reader Service Number 234.

Nutrition And The Apple

The Center for Science in the Public Interest, a nonprofit consumer group, has developed *Nutrition Express.* This game teaches the basic concepts of nutrition and diet through a series of questions and clues. Action takes place in the land of FodaFoda, where the student answers questions correctly in order to earn currency for the grocery store and to invite friends from Foda-Foda back home. The game is geared toward youngsters aged nine and up.

Nutrition Express comes with a user's guide, teaching suggestions, and a "Nutrition Scoreboard" wall chart. For the Apple II series; price is $39.95.

Center for Science in the Public Interest, 1501 Sixteenth St., NW, Washington, DC 20036.

Circle Reader Service Number 235.

Titling Videos

A new program from Videoware can put titles, custom messages, colored screens, and leaders onto videotapes. *Video Title Editor* offers a menu of more than 20 different displays, including some for weddings, birthdays, and video mail. Also included are displays for Presented By, Starring, and Credits.

The program requires a videocassette recorder and either an Apple II, Atari, Commodore 64, VIC-20, or IBM PC/PCjr. Price is $29.95.

Videoware, 19777 W. 12 Mile Rd., Suite 180, Southfield, MI 48076.

Circle Reader Service Number 236.

New IBM Telecommunications Utility

Mastercom, a new release from The Software Store, is a full-featured smart terminal and file transfer utility for the IBM PC and PCjr. It turns the computer into a terminal on a time-sharing system, captures data onto a disk and/or printer from almost any computer, and can send files to almost any type of computer. *Mastercom* supports most communication protocols including Christensen XMODEM, xon/xoff, line at a time, and no protocol. Other features of *Mastercom* include auto dial, auto answer, batch file transfer, and host mode unattended operation.

For the IBM PC, PCjr, and most compatibles. Suggested retail price is $49.

The Software Store, 706 Chippewa Square, Marquette, MI 49855.

Circle Reader Service Number 237.

Multi-Color Printing Package

A black-and-white printer can now make up to 80 full-color prints using an Apple computer and *Prince*, a new program from Baudville. The program's library of fonts can be used to make color T-shirt transfers, banners, letterheads, and labels. *Prince* can also capture any standard or double hi-res picture for editing and printing.

Four color ribbons are included for the following printers: Imagewriter, DMP, C. Itoh 8510/Prowriter, 8510 SC, NEC 8023, and Epson MX-80, RX-80, and FX-80. *Prince* sells for $69.95.

Baudville, 1001 Medical Park Dr. SE, Grand Rapids, MI 49506.

Circle Reader Service Number 238.

Electronic Trivia

Mentor Learning Systems has introduced *Ultimate Trivia*, a game featuring 4,000 facts and 200 color graphics. The facts are divided into nine categories: Music, Cinema, Geography, Sports, General Information, People, Art, History, and Television. The graphics are revealed piece by piece as each category is answered correctly. *Ultimate Trivia* can be played individually or in teams.

The program sells for $49.95 and runs on all Apple computers (with at least 64K RAM) and the IBM PC/PCjr.

Mentory Learning Systems, Inc., 1825 De La Cruz Blvd., Santa Clara, CA 95050.

Circle Reader Service Number 239.

New Infocom Adventures

Infocom has added two new products to its interactive fiction line. In *A Mind Forever Voyaging*, you play the role of a computer that has been raised as a human being up to the age of 20. You must enter a simulation of the future to see whether a plan proposed by current government and industry leaders will be beneficial for the country. Due to Infocom's new development system and an expanded 1,700-word vocabulary, the emphasis in this game is less on solving puzzles and more on revealing the story's details. (Requires at least 128K RAM; Apple II series, IBM PC/PCjr/XT/AT, Atari ST, Amiga, and Macintosh; $44.95.)

Spellbreaker completes Infocom's *Enchanter* trilogy of adventures. When a world based on sorcery finds its magic failing, you, as the leader of the Circle of Enchanters, must find and destroy the cause of this failure. (Apple II series, Amiga, IBM PC/PCjr/XT/AT, Macintosh, and MS-DOS compatibles, $49.95; Atari 400/800, XL/XE, ST, Commodore 64/128, $44.95.)

Infocom, Inc., 125 Cambridge Park Drive, Cambridge, MA 02140.

Circle Reader Service Number 240.

Parallel Printer Converters

Two new serial-to-parallel printer converters have been released by Practical Peripherals. The *Switchport IIc* was designed especially for the Apple IIc and allows the computer to be interfaced with a parallel printer. The *Switchport 232* transforms serial data into parallel, allowing any RS-232 computer to be interfaced with a Centronics parallel printer.

Both units come with a five-year limited warranty and retail for $109.

Practical Peripherals, 31245 LaBaya Drive, Westlake Village, CA 91362.

Circle Reader Service Number 241.

Productivity, Education, Entertainment Software

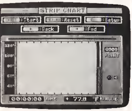
Brøderbund Software has introduced a hardware/software combination that turns your home computer into a science lab. The *Science Toolkit Master Module* includes a temperature-sensing probe, a light-sensing probe, and a special interface that connects them to an Apple II via the joystick port. Using the software's thermometer, light meter, timer, and strip chart, you can perform a wide variety of scientific experiments. (At least 64K RAM required; Apple IIe/IIc, II+ with joystick port adapter; $59.95.)

Two new packages have been added to the *Bank Street* series of productivity software. *Bank Street Mailer* is a combination letter-writing/mailing list program. *Bank Street Filer* is a database manager/report-generating program. There are two versions of each program: a 64K version for the Apple II+ and IIe offers a 40-column screen display, and a 128K version for the Apple IIc and 128K IIe offers a 40- or 80-column display and includes an onscreen calculator. They are compatible with the *Bank Street Writer* word processor. All retail for $69.95 each.

Captain Goodnight and the Islands of Fear is an arcade game that plays like an adventure movie. In your role as

Captain Goodnight, you must pilot helicopters, airplanes, tanks, trucks, and a submarine in your attempt to save the world from destruction. (Apple II series with at least 48K RAM; $34.95.)

Brøderbund Software, 17 Paul Drive, San Rafael, CA 94903.



A strip chart from Brøderbund's Science Toolkit.

Circle Reader Service Number 242.

Computerized Diet Plan From Bantam

The Complete Scarsdale Medical Diet, based on the bestselling book by the same name, is now available in a software package from Bantam Electronic Publishing. Based on sound nutritional principles, the program offers healthy, controlled weight loss. *Diet* features a meal-planning calendar, shopping list, expandable food directory, meal planner and analysis, and comparison charts.

Available for the Apple II series and IBM PC/PCjr, the program retails for $39.95.

Bantam Electronic Publishing, 666-5th Ave., New York, NY 10103.

Circle Reader Service Number 243.

Koalapad+ For Apple II

Koala Technologies has announced an enhanced version of the Koalapad, called *Koalapad+*, for the Apple IIc and IIe. The new version offers enhanced product styling, a gridded tablet surface, and additional graphics software. The software, *Graphics Exhibitor,* lets users edit images they have created.

Suggested retail price for the *Koalapad+* is $125.

Koala Technologies, 2065 Junction Ave., San Jose, CA 95131.

Circle Reader Service Number 244.

Productivity Software For Commodore 64

Datamost has announced the *KWIK* line of home productivity software for the Commodore 64. Each package includes *KWIK-LOAD!* (a Datamost fast-loading program) and retails for $19.95.

The series includes *KWIK-WRITE!,* a word processor; *KWIK-SPELL!,* a spelling checker; *KWIK-FILE!,* a database manager; *KWIK-CALC!,* a spreadsheet program; *KWIK-PAINT!,* a graphics editor; *KWIK-CHECK!,* a checkbook balancing and maintenance program; *KWIK-PAD!,* a desk secretary program; and *KWIK-PHONE,* a communications program.

Datamost, 19821 Nordhoff Street, Northridge, CA 91324.

Circle Reader Service Number 245.

Foreign Language Vocabulary Programs

Gessler Educational Software has produced three foreign language versions of its bestselling vocabulary program *Word Attack!. Bataille De Mots* (French), *Batalla De Palabras* (Spanish), and *Wortgefecht* (German) are available for the Apple II series, IBM PC/PCjr, and Commodore 64 for $49.95.

Word Attack!, as well as its foreign language versions, teaches vocabulary words and grammar with word displays, quizzes, sentence completion, and an arcade game.

Gessler Educational Software, 900 Broadway, New York, NY 10003.

Circle Reader Service Number 246.

Inexpensive Accounting Software

DAC-Easy Accounting, from DAC Software, is a seven-in-one accounting package offered at a special introductory price of $49.95. Its seven individual modules—general ledger, accounts receivable, accounts payable, billing, purchase order, inventory, and forecasting—are integrated, allowing automatic posting between modules.

The system also has spreadsheet capability, letting the user experiment with "what-if" scenarios without entering actual data. It is compatible with the IBM PC and PCjr.

DAC Software, Inc., 5580 Peterson, Suite 130, Dallas, TX 75240.

Circle Reader Service Number 247.

MIDI Editor

RolandCorp has released *MUSE (MIDI Users Sequencer/Editor)* for the Apple II series and Commodore 64. The program features eight independent tracks for recording and overdubbing sequences, track merging capability, track muting, looping by song or track length, and selectable time signatures. The editing functions can be used to insert, delete, move, copy, and rearrange measures of any track so that a composition can be changed after it has been recorded.

MUSE is compatible with any MIDI instrument and can be synchronized with drum machines, other sequencers, and multitrack tape decks. An interface is required. Suggested retail price is $150 for each version.

RolandCorp US, 7200 Dominion Circle, Los Angeles, CA 90040.

Circle Reader Service Number 248.

New PCjr Drive

A second disk drive can now be added to the IBM PCjr without adding extra circuitry or another power supply. The Junior Drive II System, from PC Enterprises, includes a 360K double-sided double-density 5-1/4 inch floppy disk drive with power supply, an adapter module, a software patch, a two-drive signal cable, and an instruction manual. The system is compatible with existing external modems, parallel printer ports, and memory expansions.

The Junior Drive II System lists for $395. For those who wish to connect their own IBM-compatible drive, the adapter module and software patch are available separately.

PC Enterprises, P.O. Box 292, Belmar, NJ 07719.

Circle Reader Service Number 249.

The Smoking Decision

A new program from Sunburst was created to alert students to the dangers of cigarette smoking. It begins by presenting facts about health risks related to smoking, and then explores issues such as peer pressure. Throughout the program, students are confronted with a series of incremental decisions, leading to a final decision whether to smoke.

The Smoking Decision is suitable for youngsters in grades 6 through 12. It runs on any Apple II computer with at least 48K memory. The $59 retail price includes a backup disk and teacher's guide.

Sunburst Communications, Inc. 39 Washington Ave., Pleasantville, NY 10570.

Circle Reader Service Number 250.

Arcade And Adventure Games For Commodore 64

Artworx has released two new games for the Commodore 64 and 128. *Falcon Patrol II* puts the user in the pilot's seat of a Falcon fighter, fully equipped with air-to-ground and air-to-surface missiles. The object of the game is to ward off the enemy's helicopter attack squadrons. Its 16 levels of play are enhanced by 3-D graphics and sound effects.

In *Sorcery,* you are the last of the great sorcerers, given new strength and powerful spells. You must use them to regain your conquered homeland and restore its previous quality of life. *Sorcery* resembles an arcade game, but

plays much like an adventure game.

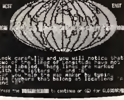
Both games retail for $19.95.

Artworx Software Company, Inc., 150 N. Main St., Fairport, NY 14450.

Circle Reader Service Number 251.

Educational Software For The Classroom

Focus Media, Inc. publishes an extensive line of classroom programs for a variety of computers. In *Za-Zoom, The Geography Genie*, students take the role of explorer as they try to determine where they are by examining the culture around them. The two programs in this package, *Travels with Za-Zoom: The World* and *Travels with Za-Zoom: The U.S.* retail for $129; either program can be purchased separately for $79. (Apple II series, Commodore 64, IBM PC/PCjr.)



Students learn about such concepts as latitude and longitude with The Language of Maps.

Students can go back in time with *The Time Tunnel: America Series Package*. During each journey, students must use clues to gather facts and guess the identities of historical figures. The package contains six programs: *Early America* (2), *A Nation Emerges* (2), and *The Presidents* (2). Suggested retail price for the complete package is $179; each series can be purchased individually for $79. (Apple II series, Commodore 64, IBM PC/PCjr.)

The Language of Maps is a series of six programs that helps students learn about maps and map terminology. Topics covered include oceans and continents; land areas and water bodies; highlands and lowlands; and finding places on maps. The Instant Computerized Glossary explains unfamiliar terms. *The Surface of the Earth* and *Location and Distance* retail as a package for $159; individually, each costs $79. (Apple II series.)

A Teacher's Lesson Planner and free backup disks are included with all packages.

Focus Media, Inc., 839 Stewart Ave., Garden City, NY 11530.

Circle Reader Service Number 252.

DISK WORLD! is proud to introduce the lowest-priced, LIFETIME-WARRANTY diskettes ever!
And they're BRAND NAME PRODUCT to boot!

5.25″SSDD→.69 ea. 5.25″ DSDD→.79 ea.
5.25″ DSDD-HD→$2.25 ea.
3.50″ SSDD→$2.09 ea. 3.50″ DSDD→$2.55 ea.

Based on multiples of 100 each.
Boxed in 10's with heavy-duty cardboard sleeves, user ID labels, reinforced hubs (where appropriate) and write-protect tabs.

Introducing Wabash Pinnacle Series Diskettes.

Two years ago, if you d told me I d be writing this ad, I would have laughed

At that time, Wabash diskettes were synonymous with "s—t"

Just saying that quality control was poor would be charitable

So much was wrong that DISK WORLO wouldn't sell them

That was yesterday.

Kearney-National Inc , a $202-million division of a much larger company, came into Wabash

Out went the old management, the old methods, the old production techniques and in went a lot of new people, ideas production lines and some really imaginative thinking

The end result.

Today, I'm proud to offer you the Wabash Pinnacle Series of diskettes at the prices shown

This isn't evolution in diskette manufacturing it's revolution

Here's what you get.

Wabash Pinnacle diskettes are

...certified 100% Error Free

...are coverd by a LIFETIME WARRANTY

meet or exceed all industry specifications (by quite some distance)

and are simply the best value in diskettes available today

The torture test.

Considering Wabash's earlier dubious reputation, I wasn't exactly a true believer when their Director of Marketing came into my office with samples

So I took a box at random selected a disk, bent the thing every which way and slipped it into my IBM-PC

It formatted It booted It stored and retrieved data

That wasn't enough.

I gave samples of the diskettes to Curt Rostenbach and, in turn, to Tom Streit, both hackers of long experience and members of the Waukegan (Illinois) Apple Users Group

Tom really went at it.

He took a quartz-halogen lamp, aimed it at the diskette until it started to smoke (and melt) and then formatted, booted the diskette and stored and retrieved data!

The same terribly (and intentionally) mutilated diskette ran on an ITT, Corona and IBM

Curt was nicer

He simply bent the diskette every which way and it still formatted booted and ran on his Apple

The best buy I've ever seen.

OISK WORLO! Inc sells more flexible magnetic media by mail-order than anyone else in the world

I as President of the corporation, won't tolerate a product with a failure rate of more than 1/1000th of 1 percent

I also don't like companies who try to milk a "quality" or "premium" image for a higher price like Oysan and Verbatim did until they failed

As President of DISK WORLD! Inc my motto is simple "the best diskette for the least amount of money

Wabash is it.

Right now, there is no better value than the Wabash Pinnacle Series of diskettes

Granted you have to buy a hundred at a time but so what? Split the order with friends, relatives, co-workers or even your worst enemies

The key thing is to get the most diskette for the money

And this is it

(Incidentally, as a corporation we put our money where our mouth is Our first order for Wabash Pinnacle Oiskettes was 1 5-million units)

That s an awful lot of faith and confidence

But then again I have the diskette that Tom Streit literally melted and kept on running

The truth about $1.00 or less diskettes.

More and more ads are popping up offering diskettes for $1 00 or less

By the same token, more and more people who were selling used cars a few months ago are now selling diskettes by mail

We did a little survey of current ads for diskettes advertised for a dollar or less and did some analysis of the market and here s what we found as it applies to 5 25″ OSOO diskettes "supposedly' selling for a dollar or less

VENDOR	ADVERTISEO LOW PRICE	ACTUAL PRICE PER 100	ACTUAL MFGR.
Unitech	89 ea	92 ea	Unspecified
Oatatech	99 ea	99 ea	Unspecified
Computer Club	95 ea	98 ea	Unspecified
	99 ea	1 02 ea	Unspecified
Communications & Electronics	49 ea	80 ea	Unspecified
Precision Data	89 ea	93 ea	Unspecified
Oiskette Connec	93 ea	93 ea	Unspecified
Comp Soft Serv	77 ea	77 ea + shpg	Unspecified
Computer/Computer	99 ea	99 ea	Unspecified
OISK WORLO	89 ea	92 ea	Wabash Datatech

The real truth about $1.00 or less diskettes.

It costs all diskette manufacturers about the same to produce a diskette Some may charge more because they want to project a "premium quality image ala the late lamented Dysan who bought their basic media from 3M

Some charge less because they sell a sub-standard product and we re not foolish enough to name names here

But here s the truth about the $1 00 or less diskette market.

It falls into four categories

1 The OISK WORLD s of the universe who simply are so big that they can buy first quality product in massive quantities and choose to pass on the savings to you (Precision Oata and Oiskette Connection on **BRAND NAME** products also fall into this category.)

2 The people who buy 'cosmos' stuff from major manufacturers that usually hits quality control standards but is cosmetically blemished and thus can't be packaged and sold under the manufacturer s own name

3 ' Ouplicator Quality Uncertified media usually below manufacturer s own standards and frequently below ANSI and IBM standards Sold on an as-is ' basis with the understanding that the manufacturer s name will never be divulged Usually about a 20% reject rate as compared to DISK WORLO s standard of less than 1/1000th of 1% reject/return rate Next to garbage this is the source of most diskettes advertised at a dollar or less

They may work and then again they may not (Frankly the odds at the Blackjack table in Las Vegas are more in your favor.)

4 Garbage Stuff that shouldn t be sold at all But some manufacturers are hurting for cash so they sell it anyway (After all they want to meet their payroll Look what happens when you don t you become a Oysan or Verbatim Lots of history, but no money) More and more garbage is being dumped into the market as manufacturers become pressed for cash and are motivated into selling anything and everything they can manufacture (Read the article in FORBES about Verbatim and its "Bonus" brand)

Finally the Taiwanese counterfeiters are moving into the act Perfect duplicates of the packaging of major manufacturers with one exception the quality isn t there

The Critical Factor.

Only DISK WORLO!, Inc offers fully brand-identified, LIFETIME WARRANTY product for less than a dollar

Every one else offering 5 25″ product for less than a buck doesn t tell you who makes it.

We do

And that ought to tell you a lot right there

Ordering & Shipping Instructions

SHIPPING Wabash Pinnacle Diskettes are sold in multiples of 100 only Shipping charges are $3 00 per 100, regardless of type or size

PAYMENT: VISA MASTERCARO and PREPAIO orders accepted Corporations rated 3A2 or better and government and quasi-government open accounts are accepted on a NET 15 basis

C O D orders are subject to a $5 00 special handling charge (Sorry for the increase but too many people have been refusing C D D orders or using bad checks It s a classic example of a few "bad eggs" making life more expensive for everyone else.)

APO, FPO, AK, HI & PR ORDERS. Include shipping as shown **and** an additional 5% of the total amount of the order to cover PAL and insurance

No other non-continental U S orders are accepted

TAXES Illinois residents only add 7%

MINIMUM ORDER $35 00

All orders subject to acceptance
Not responsible for typographical errors

ORDERS ONLY
1-800-621-6827
(In Illinois 1-312-256-7140)

INQUIRIES & INFORMATION
1-312-256-7140

FOR FASTEST SERVICE, USE MCI MAIL:
Just address **"DISKORDER"**
(24-hour shipping on any item in stock if you order via MCI MAIL.)

DISK WORLD!, Inc.

629 Green Bay Road
Wilmette, Illinois 60091

COMPUTE! Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE! Magazine
P.O. Box 10955
Des Moines, IA 50950

or call the Toll Free number listed below.

Change Of Address. Please allow us 6–8 weeks to effect the change; send your current mailing label along with your new address.

Renewal. Should you wish to renew your **COMPUTE!** subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below.

New Subscription. A one year (12 month) US subscription to **COMPUTE!** is $24.00 (2 years, $45.00; 3 years, $65.00. For subscription rates outside the US, see staff page). Send us your name and address or call the Toll Free number listed below.

Delivery Problems. If you receive duplicate issues of **COMPUTE!**, if you experience late delivery or if you have problems with your subscription, please call the Toll Free number listed below.

COMPUTE!
800-334-0868
In NC 919-275-9809

What the world really needs is a 69 cent Double Sided, Double Density Diskette with a LIFETIME WARRANTY!

And DISK WORLD! has it.

Introducing Super Star Diskettes: the high quality diskette with the lowest price and the best LIFETIME WARRANTY!

In the course of selling more than a million diskettes every month, we ve learned something higher prices don t necessarily mean higher quality

In fact we've found that a good diskette manufacturer simply manufactures a good diskette no matter what they charge for it (By way of example consider that none of the brands that we carry has a return rate of greater than 1/1,000th of 1 percent!)

In other words when people buy a more expensive diskette, they aren't necessarily buying higher quality

The extra money might be going toward flashier advertising, snazzier packaging or simply higher profits

But the extra money in a higher price isn't buying better quality

All of the good manufacturers put out a good diskette Period

How to cut diskette prices ...without cutting quality.

Now this discovery posed a dilemma how to cut the price of diskettes without lowering the quality

There are about 85 companies claiming to be "diskette" manufacturers

Trouble is, most of them aren't manufacturers

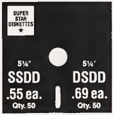
Rather they are fabricators or marketers, taking other company's components, possibly doing one or more steps of the processing themselves and pasting their labels on the finished product

The new Eastman Kodak diskettes, for example, are one of these So are IBM 5¼" diskettes Same for DYSAN, Polaroid and many, many other familiar diskette brand names Each of these diskettes is manufactured in whole or in part by another company!

So, we decided to act just like the big guys That's how we would cut diskette prices... without lowering the quality

We would go out and find smaller companies to manufacture a diskette to our specifications... specifications which are higher than most... and simply create our own "name brand" diskette

Name brand diskettes that offered high quality at low prices



Super Star diskettes are sold in multiples of 50 only Diskettes are shipped with white Tyvec sleeves reinforced hubs user ID labels and write-protect tabs

Boy, did we get lucky. Our Super Star Diskettes are the same ones you've been using for years...without knowing it.

In our search for the low priced high quality diskette of our dreams, we found something even more interesting

We found that there are several manufacturers who don't give a hoot about the consumer market for their diskettes They don t spend millions of dollars in advertising trying to get you the computer user, to use their diskettes

Instead, they concentrate their efforts on turning out the highest quality diskettes they can because they sell them to the software publishers computer manufacturers and other folks who (in turn) put their name on them... and sell them for much higher prices to you!

After all, when a software publisher or computer manufacturer or diskette marketer puts their name on a diskette they want it to work time after time everytime (Especially software publishers who have the nasty habit of copy-protecting their originals!)

Super Star Diskettes. You already know how good they are. Now you can buy them...cheap.

Well that s the story

Super Star diskettes don't roll off the boat from Pago-Pago or emerge from a basement plant just east of Nowhere

Super Star diskettes have been around for years... and you ve used them for years as copy-protected software originals, unprotected originals Sometimes depending on which computer you own the system master may have been on a Super Star diskette And maybe more than once, you've bought a box or two or more of Super Star diskettes without knowing it They just had some "big" company's name on them

Super Star Oiskettes are good So good that a lot of major software publishers, computer manufacturers and other diskette marketers buy them in the tens or hundreds of thousands

We buy them in the millions

And than we sell them to you

Cheap

When every little bit counts, it's Super Star Diskettes.

You ve used them a hundred times... under different names

Now you can buy the real McCoy the same diskette that major software publishers, computer manufacturers and diskette marketers buy... and call their own

We simply charge less

Super Special!

Order 50 Super Star Diskettes and we'll be happy to sell you an Amaray Media-Mate 50 for only $8 75, shipping included a lot less than the suggested retail price of $15 95

Regular DISK WDRLD! price: $9.69 ea. + $2 DD Shpng.

The Super Star LIFETIME WARRANTY!

Super Star Oiskettes are unconditionally warranted against defects in original material and workmanship so long as owned by the original purchaser Returns are simple just send the defective diskettes with proof of purchase postage-paid by you with a short explanation of the problem, and we'll send you the replacements (Incidentally coffee stained diskettes and diskettes with staples driven through them don t qualify as 'defective')

WE WILL MEET OR BEAT ANY NATIONALLY ADVERTISED PRICE DN THE SAME PROOUCTS AND QUANTITIES SUBJECT TD THE SAME TERMS ANO CONOITIDNS

DISKETTE STORAGE CASES

DISK CADDIES

The original flip-up holder for 10 5¼" diskettes Beige or Grey only

$1.65 ea. + .20 Shpng.

DISKETTE 70 STDRAGE

Dust-free storage for 70 5¼" diskettes Six dividers included An excellent value

$9.95 ea. +$3.00 Shpng.

HOW TO ORDER:

ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INQUIRIES:
1-312-256-7140

FOR FASTEST SERVICE, USE ND-COST MCI MAIL Our address is DISKORDER It s a FREE MCI MAIL letter No charge to you (Situation permitting we ll ship these orders in 24 hours or less)

SHIPPING 5¼" & 3½" DISKETTES—Add $3 00 per each 100 or fewer diskettes **OTHER ITEMS** Add shipping charges as shown in addition to other shipping charges **PAYMENT** VISA, MASTERCARO and Prepaid orders accepted **COD ORDERS** Add additional $5.00 special handling charge **APO FPO, AK, HI & PR OROERS:** Include shipping charges as shown and additional 5% of total order amount to cover PAL and insurance We ship only to United States addresses except for those listed above **TAXES** Illinois residents add 7% sales tax

MINIMUM ORDER: $35 00

HDURS
Human: 8AM-6PM Central Time, Monday through Friday
Answering Machine 6PM-8AM, All Times
MCI MAIL: 24 hours a day.

DISK WORLD!, INC.

629 Green Bay Road
Wilmette, Illinois 60091

Incredible value!

Nashua™ Diskettes

LIFETIME WARRANTY!

85¢ ea. 5¼" SSDD Qty 50 — **97¢ ea.** 5¼" DSDD Qty 50

These are poly-bagged diskettes packaged with Tyvek sleeves, reinforced hubs, user identification labels and write protect tabs. NASHUA Corporation is a half-billion dollar corporation and a recognized leader in magnetic media.

SOFT SECTOR ONLY! Sold in multiples of 50 only!

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois 1-312-256-7140)
INFORMATION & INQUIRIES: 1-312-256-7140

HOURS 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette Illinois 60091

DISK WORLD! Authorized Distributor NASHUA MAGNETIC MEDIA

FANTASTIC LOW PRICES ON

BASF QUALIMETRIC DISKETTES!

LIFETIME WARRANTY!

Buy in bulk and save. 150 to the carton with envelopes, write protect tabs and user ID labels. Boxed product is the same except in cardboard boxes of 10.

	Qty 50	Qty 150
5.25" SSDD	.83 ea	.80 ea
5.25" DSDD	.94 ea	.92 ea
5.25" DSDD-HD	2.13 ea	N/A
5.25" SSDD-96TPI	.94 ea	N/A
5.25" DSDD-96TPI	1.06 ea	N/A
3.50 SSDD-135TPI	1.84 ea	1.68 ea
3.50 DSDD-135TPI	2.40 ea	2.28 ea

NOTE: 3.50" diskettes in Quantity 50 are packed in plastic library cases. That's why they seem to be a better buy. But there are only 5 diskettes to a case, so the bulk diskettes are really a better deal unless you like expensive little library cases.

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois 1-312-256-7140)
INFORMATION & INQUIRIES: 1-312-256-7140

HOURS 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD! Authorized Reseller Information Processing Media BASF

Now, the lowest prices ever on

3M Scotch® DISKETTES

LIFETIME WARRANTY!

The best deal on 3M diskettes you've ever seen!

FREE!

FLIP 'N FILE 15 with every order for 5.25" SSDD and DSDD.

$1.36 ea. 5.25" SSDD with FREE Flip 'n File 15
5.25" DSDD with FREE Flip 'n File 15 **$1.67 ea.**
MINIMUM ORDER 50 Diskettes
(Orders less than 50 are $1.39 and $1.70, respectively.)

100% LIFETIME WARRANTY!

This is a Super Special Promotion. It was supposed to end May 31, 1985.

But we decided to buy more than 1,000,000 3M diskettes packed in the FREE Flip 'n File 15... and give you the benefits of this terrific value.

One word of warning: this offer is limited only to supplies on hand. Once this inventory is gone, that's it. The prices stay the same... but there's no FREE Flip 'n File.

The last time we ran an offer like this, everything sold out in about six weeks.

So don't wait. Order now.

Other 3M diskettes

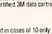
	Qty 20	Qty 50
5.25" SSDD-96TPI	$2.07 ea	$2.04 ea
5.25" DSDD-96TPI	$2.51 ea	$2.48 ea
5.25" DSDD-HD FOR AT	$3.18 ea	$3.14 ea
3.50"SSDD-135TPI(Mac)	$2.29 ea	$2.26 ea

DATA CARTRIDGES

100% certified 3M data cartridges

DC-1000	$12.62 ea
DC-300XLP	$19.38 ea
DC-600A	$20.61 ea

Sold in cases of 10 only
Add $5.00 shipping per 10



FIRST TOUCH KEYBOARD STRIP

The best way we've ever seen to avoid static danger. Adheres to the front ledge of your keyboard. Overpriced at $19.95 so we cut the price to $10.95 + $1.50 Shipping.

A VERY SPECIAL OFFER
3M POST-IT NOTES & DISPENSER

Suggested Retail: $20.95. Includes dispenser, 12 pads of 3" x 5" Post-It notes and Pilot Ball Point Pen. Order it with 50 3M diskettes or 10 data cartridges and it's only $9.75 + $3.00 Shipping.

DISK WORLD! Authorized Distributor Information Processing Products 3M

ATHANA DISKETTES

The great unknown!

79¢ ea 5¼" SSDD Qty 50 — **85¢ ea** 5¼" DSDD Qty 50

You've used these diskettes hundreds of times... as copy-protected originals on some of the most popular software packages. They're packed in poly-bags of 25 with Tyvek sleeves, reinforced hubs, user identification labels and write-protect tabs.

LIFETIME WARRANTY!

SOFT SECTOR ONLY! Sold in multiples of 50 only

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois 1-312-256-7140)
INFORMATION & INQUIRIES: 1-312-256-7140

HOURS 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette Illinois 60091

DISK WORLD! Authorized Distributor ATHANA MAGNETIC MEDIA

DISK WORLD!

Ordering & Shipping Instructions

Shipping: 5¼" & 3.5" DISKETTES—Add $3.00 per each 100 or fewer diskettes. **Other Items:** Add shipping charges as shown in addition to other shipping charges. **Payment:** VISA and MASTERCARD accepted. **COD Orders:** Add additional $5.00 Special Handling charge. **APO, FPO, AK, HI & PR Orders:** Include shipping charges as shown and additional 5% of total order amount to cover PAL and insurance. **Taxes:** Illinois residents only add 7% sales tax.

Prices subject to change without notice.
This ad supercedes all other ads.
Not responsible for typographical errors.
MINIMUM TOTAL ORDER: $35.00

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois 1-312-256-7140)
INFORMATION & INQUIRIES: 1-312-256-7140

HOURS 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette Illinois 60091

DISK WORLD!

DISKETTE STORAGE CASES

AMARAY MEDIA-MATE 50: A REVOLUTION IN DISKETTE STORAGE

Every once in a while someone takes the simple and makes it elegant! This unit holds 50 5¼" diskettes, has grooves for easy stacking, inside nipples to keep diskettes from slipping and several other features. We like it! **$9.69 ea** + $2.00 Shpng

DISKETTE 70 STORAGE: STILL A GREAT BUY.

Just free storage for 70 5¼" diskettes. Six dividers included. An excellent value. **$9.95** + $3.00 Shpng

DISK CADDIES

The original flip-up holder for 10 5¼" diskettes. Beige or grey only. **$1.65 ea** + 20¢ Shpng

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois 1-312-256-7140)
INFORMATION & INQUIRIES: 1-312-256-7140

HOURS 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette Illinois 60091

DISK WORLD! The value leader in Computer supplies And accessories.

PRINTER RIBBONS:

at extraordinary prices!

Brand new ribbons manufactured to Original Equipment Manufacturer's specifications in housings. (Not re-inked or spools only.)

LIFETIME WARRANTY!

Epson MX-70/80	**$2.70** ea + 25¢ Shpng
Epson MX-100	**$4.08** ea + 25¢ Shpng
Okidata Micro83	**$1.37** ea + 25¢ Shpng
Okidata Micro84	**$2.98** ea + 25¢ Shpng

FOR ORDERS ONLY: 1-800-621-6827 (In Illinois 1-312-256-7140)
INFORMATION & INQUIRIES: 1-312-256-7140

HOURS 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette Illinois 60091

DISK WORLD!

MUST LIQUIDATE At Far Below Dealer Cost!
TOTAL Personal Computer System

Factory Reconditioned with Factory Warranty!

Carrias easily as a sultcase! Plugs into 11SV outlat!

GREAT GIFT IDEA FOR STUDENTS!

Sorry, we're not permitted to PRINT the famous brand-name. BUT, we CAN "tell all" if you call us TOLL FREE: 1-800-328-0609!

THE COMPUTER

Snap-on computer keyboard! 64K RAM, 20K ROM. Full-size typewriter keyboard. Upper and lower case letters, numerals, symbols, reverse characters. 2 cursor control keys, 4 function keys, programmable to 8. Music synthesizer with 3 independent voices, each with 9 octave range. Input/output ports accommodate . . . user, serial, ROM cartridge, joysticks, external monitor, phone modem.

Built-in disk drive! Intelligent high speed unit with 5¼" floppy disk recorder. 170K formatted data storage, 35 tracks. 16K ROM. Uses single sided, single density disk. Serial interface. Second serial port to chain second drive or printer.

Built-in color monitor! Displays 40 columns x 25 lines of text on 5" screen. High resolution 320 x 200 pixels. 16 background, character colors.

Built-in ROM cartridge port! Insert ROM program cartridge. Multitude of subjects available in stores across the nation!

Original List Price $995.00

Liquidation Priced At Only $388

Item H-725-63631-00 Ship, handling: $20.00

THE PRINTER

Print method: Bi-directional impact dot matrix

Character matrix: 6 x 7 dot matrix

Characters: Upper and lower case letters, numerals and symbols. All PET graphic characters.

Graphics: 7 vertical dots — maximum 480 columns. Dot addressable.

Character codes: CBM ASCII code

Print speed: 60 characters per second

Maximum columns: 80 columns

Character spacing: 10 characters per inch

Line feed spacing: 6 lines per inch in character mode or 8 lines per inch selectable. 9 lines per inch in graphics mode.

Line feed speed: 5 lines per second in character mode. 7.5 lines per second in graphics mode.

Paper feed: Friction feed

Paper width: 4.5" to 8.5" width

Multiple copies: Original plus maximum of two copies.

Dimensions: 13"W x 8"D x 3¼"H. Wt.: 6½ lbs. **Power:** 120V AC, 60 Hz.

Original List Price: $200.00

Liquidation Priced At $118

Item H-725-63831-00 Ship, handling: $7.00

THE SOFTWARE

"Easy Script" One of the most powerful word processors at any price! Cut re-typing, create documents from standard paragraphs, do personalized letters, see and change a document before it is printed. Instruction manual has extensive training section that simplifies use . . . even for someone who has never used a computer or word processor before!

"The Manager" A sophisticated database manager for business or home use. Business uses: accounts payable/receivable, inventory, appointments, task manager. Home uses: mailing lists, home inventory, recipes, collection organizer, investment tracking, checkbook balancing. School uses: research article index, gradebook.

Original List Price: $73.98

Liquidation Price $24

Item H-725-64011-03 Ship, handling: $3.00

BUY INDIVIDUAL UNITS OR GET THIS ULTRA-FAMOUS SYSTEM AT ONE LOW PACKAGE PRICE!

TOTAL Personal Computer System available at FAR BELOW dealer cost!

Original List Price $1,293.00

TOTAL PACKAGE PRICE....... $488

Item H-725-64011-02 Ship, handling: $24.00

Compatible with above Computer System (Not included in package price.)

JOYSTICKS (Set of 2)

Mfr. List: $59.90 pr.

Liquidation Price .... $18 pr.

Item H-725-63522-00 S/H: $6.00 pr.

64 MODEM

Mfr. List: $124.95

Liquidation Price $44

Item H-725-63646-00 S/H: $4.00

VISA MasterCard

Credit card members can order by phone, 24 hours a day, 7 days a week.

Toll-Free: 1-800-328-0609

TeleCheck Your check is welcome! No delays when you pay by check!

C.O.M.B. CO.®

Authorized Liquidator

14605 28TH AVENUE NORTH
MINNEAPOLIS, MINNESOTA 55441-3397

SEND ME THE ITEMS I HAVE LISTED BELOW

Price subject to change after 60 days. Sales outside continental U.S. are subject to special conditions. Please call or write to inquire.

Item No.	#	Item	Price	S/H
		TOTAL		

C.O.M.B. CO.® Item H-725
14605 28th Ave. N./Minneapolis, MN 55441-3397

Send the items indicated at left. (Minnesota residents add 6% sales tax. Please allow 3-4 weeks delivery. Sorry, no C.O.D.)

☐ My check or money order is enclosed. (No delays in processing orders paid by check, thanks to TeleCheck.)

Charge: ☐ MasterCard® ☐ VISA

Acct. No. ______ Exp. ______

PLEASE PRINT CLEARLY

Name ______

Address ______

City ______

State ______ ZIP ______

Phone ______

Sign here ______

WHITE HOUSE COMPUTER

P.O. Box 4025, Williamsport, PA 17701

"Where Prices Are Born, Not Raised"

TOLL FREE 1-800-351-3442

PA CALL 1-717-322-7700

— PA Residents FREE Shipping —

MONDAY - FRIDAY 9 AM - 6 PM
VISA & MC ACCEPTED 4%

POLICY. No deposit on COD orders. Free freight on all prepaid cash orders over $300 in the continental USA. APO & FPO add $5.00 per hundred. For priority mail add $8.00 per hundred. PA residents add 6% sales tax. Defective products must have Prior RA number. Schools net 15

PRINTERS

EPSON		**OKIDATA**	
RX 80	209.00	Okimate 10	179.95
RX 100	369.00	182	219.95
JX 80	479.00	84	640.95
FX 85	345.00	**192**	**349.95**
FX 185	499.00	193	525.95
LQ 1500 PAR	979.00	**COMMODORE**	
LQ 1500 SER	1039.00	MPS 801	165.95
LX 80	222.00	802	199.95
Homewriter	209.00	803	165.95
Comrex 220 Atari	199.00	DPS 1101	295.95
Comrex 220 Comm	199.00	**STAR MICRONICS**	
LX 90	245.00	**SG-10**	**214.00**
SQ 2000	1525.00	SG-15	379.00
DX 10 Daisywheel	235.00	SD-10	345.00
DX 20 Daisywheel	319.00	SD-15	450.00
HS 80 Letterjet	359.00	SR-10	485.00
ATARI		SR-15	585.00
XTM 201	99.95	Powertype	307.00
XTC 201	109.95	**LEGEND**	
XDM 121	209.95	**1380**	**269.00**
XMM 801	169.95	**1385**	**305.00**
STC 504	139.95	1080	229.00
STD 121	219.95	880	199.00
SMM 801	279.95	**PANASONIC**	
CITIZEN		1090	187.00
MSP 10	305.00	1091	CALL
MSP 15	450.00	1092	385.00
MSP 20	450.00	1093	425.00
MSP 25	575.00	3151	455.00

INTERFACES

850 (Atari)	108.95
UPrint/port	49.95
UPrint/16k Buffer	**69.95**
UPrint/64k Buffer	89.95
Cardco G	39.95
G-Wiz	**49.95**
Apple Dumpling GX	**59.95**

PRINTER PAPER

2500 Sheets	
Lazor Edge	**24.95**
1000 Shts Lazor	14.95
500 Shts Lazor	9.95
Color Paper	
Assorted Pastels	
2500 Shts Lazor	42.95
1000 Shts Lazor	23.95
500 Shts Lazor	14.95

SOFTWARE

Print Shop	28.95
Graphics Library I	17.50
Graphics Library II	17.50

VIDEO CASSETTE RECORDER $279.00

DISK DRIVES

ATARI	
Indus GT	**205.95**
1050	164.95
Happy 1050	335.95
Happy Enhancer	160.95
MSD Dual	459.00
C-64	
Indus GT	**229.00**
1541	195.00
1571	249.95
1572	375.95

MODEMS

C-1650	54.95
C-1670 300/1200	185.95
MPP 1064	54.95
Tele Learning	39.95
Hayes 300	149.95
Hayes 1200	385.95
Mitey Mo	59.95
Westridge	59.95
Compuserve	19.95
Micro Stuffer	95.95
MPP 1000E	**69.95**

DISKETTES

SKC	
SS/DD	10.95
DS/DD	14.95
BONUS	
SS/DD	9.50
DS/DD	13.50
MAXELL	
MD 1	15.95
MD 2	20.95

MONITORS

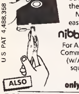
ZENITH	
122 A	82.95
123 G	75.95
AMDEK	
300 G	119.00
300 A	129.00
Color 300	**229.00**
Color 500	339.00
Color 600	399.00
Color 700	469.00
Color 710	539.00
310A	145.00
ATARI	
XC 141	199.95
SM 124	135.95
SC 1224	335.95
TEKNIKA	
MJ-10	185.95
MJ-22 RGB	**269.95**
SAKATA	
SC 100	179.00
COMMODORE	
C-1702	185.95
C-1902 RGB	259.95
C 1901	129.95

COMPUTERS

COMMODORE	
C 64	139.95
C-128	275.95
ATARI	
800 XL	89.95
130 XE	139.95
520 ST Monochrome & More	CALL
520 ST Color RGB & More	CALL

IT'S LIKE

FREE DISKETTES



Your 5¼" single side disks are usable on the other side. You paid for one side, why not use the other... **IT'S FREE!**

Nibble Notch will **open** your **new** disk. It's easy... won't harm existing data. Try it!

nibble notch I
For Apple, Franklin, Commodore & Atari (w/Atari Drives), square notch
only $14.95* PLUS P&H

nibble notch II
For all other computers, square notch & index hole
only $21.90* PLUS P&H

DISK OPTIMIZER II©

Apple II Series Software
Pro DOS • DOS 3.3 • Pascal
Examines your **new** disk, locks out bad sectors and **certifies it 100% ERROR-FREE** in 30 seconds or less! Also checks drive speed... and more!

SUPER SAVER PACKAGE
Nibble Notch I and Disk Optimizer Combo (Optimizer alone reg. $24.95)
$29.95* FOR BOTH!

QUALITY DISKETTES low as 99¢

*add $2 ($5 frgn) for P & H. Fl. Res. add 5% Sales Tax

Toll Free 1-800-642-2536

nibble notch® computer products

FL 1-305-748-3770
OR SEND CHECK OR MONEY ORDER TO

4211 NW 75th TERRACE, • DEPT. 662 LAUDERHILL, FL 33319

PROFESSIONAL HANDICAPPING SYSTEMS

• PRESENTED BY PROFESSOR JONES •

GLO. Thoroughbred "Gold" Edition™
A "Full" featured thoroughbred analysis designed for the professional and the serious novice. **$159.95** complete

EGLD. Enhanced "Gold" Edition™
"Gold" Edition with complete Master Bettor™ system integrated onto the same disk. This powerful program will transfer all horses and scores to the bet analysis with a "single keystroke." (Master Bettor™ included) **$199.95** complete

GLTO. Limited "Gold"™
Enables **Professional Handicappers** to assign specific values to the racing variables "they" feel are important. Create program weight based on a particular track and fine tune it for **maximum** win percentage. This program is designed for "ease of use." The user needs **no programming experience**.
(contains Integrated Bettor™) **$299.95** complete

GO. Gold Dog Analysis™
The "ONLY" professional greyhound analysis available that evaluates ALL variables. **$149.95** w/integrated Bettor **$199.95**
Limited Version **$299.95**

MHH. Master Harness Handicapper™
Professional software designed to provide a thorough analysis of all trotter and pacer races in North America and Canada. **$159.95** complete
$199.95 w/Integrated Bettor. Limited **$299.95**

Professor Pix Football™
Complete STATISTICAL ANALYSIS on Data Base allowing "Designated" previous games to be evaluated. Statistical Series **$99.95**
w/Win-Loss Power Ratings **$149.95**

NBA. Basketball™ NBA **$99.95** w/college **$129.95**
w/power ratings **$149.95**

LOT. Lottery Analysis™
Statistical comparison program designed to detect subtle patterns in winning numbers and digits. Lottery (3-4 digit) **$79.95**
w/Lotto (Max. 99 Digit) **$99.95**

PC-3 Portable Computer (4k) with choice of Thoroughbred, Greyhound or Trotter™ **$249.95** (Includes portable computer and program.)

M-100 Portable (24k) w/choice of Thoroughbred, Greyhound or Trotter **$649.95** (Includes portable computer and program.)

BROCHURE AVAILABLE

IBM™
APPLE™
TRS-80™
CPM™
COMMODORE™

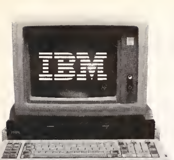
Prof. Jones
1940 W. State St.
Boise, ID 83702

VISA MasterCard

48 HR. FREE SHIPPING

CALL 208-342-6939

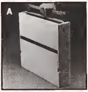
TERMS: FREE SHIPPING ALL SOFTWARE. Add $6.00 hardware / $6.00 C.O.D. / UPS Blue $6.00 / Out of Country $9.00 / ID Residents 4% / 3 weeks personal checks / Cash price only, add 2% Visa. MC / Prices subject to change

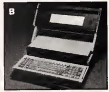


$699 Computer Breakthrough

IBM compatible computer with 128K memory and two disk drives is more than a bargain.

A *It carries like a briefcase.*
B *With LCD lid open*





The Visual Commuter sells for only $699 The IBM monochrome monitor is optional.

It's a new concept. And if my hunches are correct, the new Visual Commuter will take years for others to copy.

By Joseph Sugarman, President

First, it is a tremendous bargain. At $699 nothing even comes near. Secondly, it's powerful. You get 128K memory that's expandable to 512K. And finally, it's totally IBM compatible. It will even run the Flight Simulator program and Lotus Symphony™ But there's more.

The unit can be used as a stand-alone office computer as shown above or it can be packed up and used as a portable—but without all the weight. There's an optional 16-line x 80 character LCD display that pops up to replace the heavy CRT monitor. Unlike the smaller portable computer keyboards, the Commuter keyboard is a full-size replica of the IBM with its ten function keys and numeric key pad. And the Commuter uses 5¼" disks so you have full access to all the popular IBM software.

It's lighter (only 18 lbs), flatter (only 3½" thick) and carries like a briefcase. In short, the Visual Commuter is a combination of all the good features of a portable computer (size, weight, portability), all the good features of a traditional desktop computer (full-size keyboard, 5¼" disks, full power) and none of the disadvantages of either.

USE IBM PROGRAMS

Even if you have another computer but miss some of the IBM programs, for only $699 and an IBM compatible monitor, you've got a complete MS/DOS system.

It was also made modular so you can select just those components that you need for your particular application. For example, you may not need the 16 line x 80 column LCD display which adds 2 lbs to the unit's weight (a blank lid comes with the unit). Or you may not need e monitor because your other computer may already have one. But you may want more power—256K or 512K—so you order just what you need.

For all you technical people, listen to these specs. There's a 16-bit 8088 processor, 128K memory with parity, parallel printer port, serial ASYNC RS 232C port, Din connector RF modulator or composite video output for TV and composite video input monitors, RGB/direct drive output for high resolution monochrome or color monitors, IBM compatible color graphic support, support logic for 80x25 or 40x25 character display and LCD display, connector to IBM expansion unit, disk controller supporting two 5¼" disk drives, ANSI standard ROM-based terminal emulation, and ROM-based extended diagnostics. The dual disk drives are double sided-double density (360 Kbytes). The Commuter runs at the same clock speed as the IBM PC (4.77 MHz) but because of its new design, it runs between 8 to 10 percent faster.

ATTRACTIVE CASE

There's an attractive carrying case made by American Tourister that holds your software, your power cord, your documents and even our optional 1200 baud modem. The compatible Maxwell modem lets you communicate with other data banks. Made by the world's largest modem manufacturer, Racal-Vadic, it is normally a $500 value but our price is only $249 which includes a complete communications software package. There's also a toll free, online warranty service and a customer hotline to answer any of your technical questions.

You may have recently heard of Visual Technology Incorporated. They are innovators in the design and manufacture of smart alphanumeric terminals and some of the finest graphic terminals in the country.

The Visual Commuter was scheduled to sell for over $2500 with the LCD display. And even at that price, when compared to the IBM system, it was a good value. But JS&A and Visual (in a joint venture with SGD Holding Corp.) saw the opportunity of having just one customer. Together, by selling directly to you, we've eliminated the distributors, dealers and all the sales, administration and advertising costs and have passed the savings on to you. But there are a few catches.

JUST A FEW THOUGH

Once we install the memory, you'll have to send the unit back to us to add more memory. So we ask that you estimate, in advance, the maximum power that you'll require for your needs. 128K memory is plenty for most applications but if you want to run Lotus Symphony, you'll need all 512K. Secondly, we ask that you act quickly. Although we have most of the product in stock right now, there's always the chance that we'll run out.

The Visual Commuter measures only 3½ x 15½ x 18" wide and comes complete with power cord (it only operates on standard AC current), the operating system (Micro-Soft's MS/DOS ver. 2.1) complete with besic and utilities, two beautifully written manuals, lid (without LCD display) and a limited 90-day warranty. There are service centers throughout the United States set up to service the unit in addition to the service-by-mail facility at Visual's home office near Boston.

I urge you to give the Visual Commuter a test. Order one from JS&A and use it for 30 days without risk. Plug in your IBM monitor and load any of the IBM software you currently have. See how the large keyboard matches the IBM perfectly and how its handle makes a perfect hand rest while typing or a comfortable handle for carrying the unit. See how convenient the unit is to take home or bring with you on a trip with its fold open LCD monitor. If you don't feel that the Visual Commuter is more than you expected, pack it up and ship it back within 30 days for a prompt and courteous refund including the $25 postage charge. You can't lose.

PERSONALLY USED

I have personally used the Visual Commuter. I have taken it with me on trips, set it up as a stand alone by plugging in my IBM monitor. I have run everything from Symphony™ to Wordstar®—from 1-2-3® to the Flight Simulator program. I strongly recommend the system.

To order, send your check or money order to the address below listing the items and order numbers (shown in parentheses) or order by credit card using our toll free number below.

Thanks to the latest in technology and a direct-to-consumer marketing program, this system can be yours at a tremendous value. Order your Visual Commuter at no obligation, today.

Visual Commuter with 2 Disk Drives and 128K Memory (6105)	**$699**
16 x 80 LCD Alphanumeric Display (6106)	**299**
IBM 12" Green Monitor (6107)	**229**
Zenith 12" Amber Monitor (6114)	**99**
Expansion to 256K (6110)	**199**
Expansion to 512K (6111)	**399**
Maxwell 1200 Baud Modem (6112)	**249**
Carrying Case (6115)	**69**

Add $25 postage and handling per system. IBM is a Trademark of International Business Machines Corporation.

JS&A PRODUCTS THAT THINK

Dept. PT One JS&A Plaza
Northbrook, Il 60062
QUESTIONS ONLY: (312) 564-7000
CALL TOLL FREE 800 228-5000
IL residents add 7% sales tax ©JS&A Group, Inc.,1985

Classified

SOFTWARE

ATARIWARE FROM THE LIBRARY OF JAX
ATARI - Computer Enthusiasts! Over 70 disks to choose from. $7.95 ea. Catalog with SASE. JACE, 1187 Dunbar Ct., Orange Park, FL 32073

COMMODORE: TRY BEFORE YOU BUY. Top 25 best-selling games + classics, new releases. Visa, Mastercard. Free brochure. Rent-A-Disc, 908 9th Ave., Huntington, WV 25701. (304) 522-1665

ATARI ST USERS - Life organizer & entertainment jackpot. Big software package. Write: MC, 94 Macalester Bay, Winnipeg, Manitoba R3T 2X5 Canada

BASICally SIMPLE, a fast, easy method to learn programming in basic on your Commodore 64. Clear, concise instruction in the use of all commands. No gobblede-gook. Send $20 for this tutorial and reference disk to: Free Spirit Software, Inc., 5836 S. Mozart, Chicago, IL 60629

TI-99/4A SST Expanded Basic Compiler $49.00. Converts Basic to machine language. Requires E/A or MM, Disk, Memory Expansion. Compiler and Graphics Utility $59.00. SST SOFTWARE, P.O. Box 26, Cedarburg, WI 53012

HOW GOOD A COACH ARE YOU? CALL THE plays against Atari or a friend. Football text game features varied team strength, weather, overtime, 2 pt. conversions, team stats, standings + more! Easy to play. For Atari 800 + Basic. $25 disk. James Franklin, 1600 NE 135 St., Apt. 803, N. Miami, FL 33181

Free Educational Software Catalog
158 pp, color pictures, 1000+ programs for Com., Apple, IBM, Atari. Send name & address to: Interstate Software, P.O. Box 8952, Boise, ID 83707, (208) 342-3347

ATARI MICROSOFT Basic I/II: Try ACEY DUCEY, a great casino card game for 1-4 players. ACEY DUCEY is designed as a tutorial, utilizing Microsoft Basic's best features. Full documentation explains every detail. 400/800/XL/XE disk or tape + docum. Send $8.50. CALTECH, Box 136, Kenilworth, NJ 07033-0136

ORGANIZE YOUR GENEALOGY with BRANCHES on your C-64 & 1541 drive. Send $14.95 to Inventure Soft., 3353 South Main, Suite 101, Salt Lake City, UT 84115

TI-99/4A QUALITY SOFTWARE for Business, Home and Entertainment ** BONUS Software Offer! ** Send for FREE Catalog to MICRO-BIZ HAWAII, Box 1108, Pearl City, HI 96782

AT BAT STATS 64 - Baseball stats for C64. Keeps 22 offensive stats, holds 162 games and 40 players. Prints 7 reports. On disk $32.95 postpaid. THE 43RD ST. SOFTWARE SHOP, 720 43rd St., Ashland, KY 41101. Ideal for coach or player! (606) 324-7403

GOMUKO — Compiled basic program 15 by 15 board, Unmove, load/save game user changable logic for 64. Send $14.95 to: W. L. Schneider, 6027 S. High, Suite 410, Oklahoma City, OK 73149

IBM PCjr. OWNERS: We have the software you want. We also have kits for easy 2nd drive addition. Free catalog. OOWL SOFTWARE, 1435 Burnley Sq. N., Columbus, OH 43229

TEACHERS: ELECTRONIC GRADEBOOK
for Atari. Powerful, simple to use. Record keeping, grade averaging. Screen display and variety of pre-formatted printed reports. Clear instructions. Requires 48K and one disk drive. $29.95. GRADESCAN, 1722 Golden Ct., Crofton, MD 21114

TI-99/4A Software/Hardware bargains. Hard-to-find items. Huge selection. Fast service. Free catalog. D.E.C., Box 690, Hicksville, NY 11801

REAL ESTATE SOFTWARE for C64, Apple II, IBM-PC. Includes Mortgage Payments, Amortization, Depreciation, Cash Flows and Taxable Income Effects. All for $35 pp. J.A.L., Box 234, East Texas, PA 18046. (215) 398-0434

APPLE II SERIES, COMM 64, TI 99/4A, IBM/PC Discount Entertainment Software. Specify which free catalog. New titles. The Adventure Store, Roscoe, IL 61073-0320

TOR FIGHTER: You are entrusted with the daring adventure of defending earth, $12. C64, C16, +4 (disk), TI (cass.). Check or money order to: Computex, Box 7245, Ponce, PR 00732

TIM, The Investment Manager and COMEX Gold and Silver Management, plus Fun-Anl Analyze Stock Program for 64. All three $19.95 or write for free details to: W. L. Schneider, 6027 S. High, Suite 410, Oklahoma City, OK 73149

LOTTO PICKER. Go for Million Dollar Jackpots! Picks all USA Lotto games +! PRO FOOTBALL ANALYST. Beat the points consistently and easily! They pay for themselves! Each $35. IBM-PC/TI99 ver. RIDGE, 170 Broadway, #210-C NYC, NY 10038. 718-833-6335. Free Catalog

HARDWARE

Trade in your used Commodore or Atari on a brand new C-128 or Atari ST. This offer may not be available through retail outlets. Brochure $2.00 and SASE. NEW WEST TECHNOLOGY, 4B Monroe Pkw., Box 200, Ste. 134, Lake Oswego, OR 97034

HARDWARE & SOFTWARE 30% BELOW RETAIL. Apple, Atari, C64, IBM-PC, TI-99. Over 1000 titles. Hard to find items. Send $1.00 for catalog. Specify computer. Multi-Video, P.O. Box 246, East Amherst, NY 14051

MISCELLANEOUS

64 AUTHOR'S CLUB - We get you published. Send for free details or send $25.00 (a 50% savings) to: AUTHOR'S CLUB, 6027 S. High, Suite 410, Oklahoma City, OK 73149

1986 CALENDAR. The history of computers 1592-1985, especially micros. $6 ch or m o. Spiral, 3 holes, 8½ × 11. HEATSTROKE SOFTWARE, P.O. Box 62171, Tucson, AZ 85734-6171

HELP IS ON THE WAY!
Just call 1-800-334-0868 to get your free copy of the latest COMPUTE! Books Catalog! If you need help in getting information on all of the latest COMPUTE! book titles available plus all COMPUTE! backlist titles, call us today!

NATIONWIDE BBS DIRECTORY. 250 exciting network numbers. Computer to computer. Coast-to-coast and Canada. Send $3 to: Compuquest, Box 267, Fairfax, VA 22030-0267.

RIBBONS for ANY PRINTER at LOW PRICES!! DELTA MICRONICS
BOX 10933, ERIE, PA 16514
(814) 455-5667

*** MR. SOFTWARE CO. ALL POPULAR TITLES ***
* Printers, Monitors, Drives, VISA, MC * heavy discounts - Send $1.00 for catalog 11-9 Exton Complex, Somers Point, NJ 08244

FREE! USE YOUR MODEM! Call our innovative electronic shopping center, FANTASY PLAZA. VISA and MASTERCARD accepted. 300 BAUD. You've never seen anything like it! Use your Modem NOW! (818) 840-8066

EARN MONEY, PART OR FULL TIME, AT HOME with your computer. 50 page manual with forms. Money back guarantee, $9.95, JV Tech, P.O. Box 563, Ludington, MI 49431

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: $25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines.)
Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted. Make checks payable to COMPUTE! Publications.
Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.
General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Orders will not be acknowledged. Ad will appear in next available issue after receipt.
Closing: 10th of the third month preceding cover date (e.g., June issue closes March 10th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.
Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

maxell DISKS
LIFETIME WARRANTY

TIRED OF WAITING FOR SERVICE AND PRICE?
9 out of 10 SURVEYED DISK BUYERS PREFERRED

NORTH HILLS
#1 IN SERVICE AND PRICE
1-800-328-3472

Formatted and hard sectored disks in stock-Dealer inquiries invited.
COD, VISA, MASTERCARD
All orders shipped within 24 hrs.

NORTH HILLS CORP. INTERNATIONAL
3564 Rolling View Dr
White Bear Lake, MN 55110
MN call collect—612-770-0485

Maxell Floppy Disks

The Mini-Disks with maximum quality.

Dealer inquiries invited. C.O.D's accepted
Call FREE (800) 235-4137

PACIFIC EXCHANGES
100 Foothill Blvd. San Luis
San Luis Obispo, CA 93401
In Cal call (800) 592-5935 or
(805)543-1037

STATE-OF-THE-ART MAGNETIC MEDIA

5¼" DISKETTES

- With Hub Rings
- With Write Protect Tabs
- With Static-Free, Dust-Free Envelopes
- With User ID Labels
- In Factory Sealed Poly Packs

69¢	79¢
Single Side Double Density	Double Side Double Density

100% ERROR FREE - LIFETIME WARRANTY
MEET OR EXCEED APPLICABLE ANSI ECMA ISO STANDARDS

Minimum quantity 50 diskettes. Discount for 300 or more diskettes. Shipping and Handling $4.00 per 100 diskettes. Reduced shipping charge for larger quantities. C.O.D. add $4.00. Cash or certified check. MI residents add 4% sales tax. Prices subject to change without notice.

COD

Precision Data Products
P.O. Box 8367, Grand Rapids, MI 49508
(616) 452-3457 • Michigan 1-800-632-2468
Outside Michigan 1-800-258-0028

3M Diskettes
Lifetime Warranty

TIRED OF WAITING FOR SERVICE AND PRICE?
9 out of 10 SURVEYED DISK BUYERS PREFERRED

NORTH HILLS
#1 IN SERVICE AND PRICE
1-800-328-3472

Formatted and hard sectored disks in stock-Dealer inquiries invited.
COD, VISA, MASTERCARD
All orders shipped within 24 hrs.

NORTH HILLS CORP. INTERNATIONAL
3564 Rolling View Dr.
White Bear Lake, MN 55110
MN call collect—612-770-0485

DISKS 75¢

SSDD

QUALITY MEDIA
LIFETIME REPLACE GUARANTEE
HUB RINGS, TYV EPS.
WRITE PROTECTS

Because we buy in huge volume to supply software manufacturers, our prices can't be beat

	50+	250+	500+	1000+
5.25" SSDD	1.00	.90	.85	.80	.75
5.25" DSDD	1.15	1.05	1.00	.95	.85
PC FORMATTED	1.45	1.35	1.25	1.15	1.10
3.5" 1D (For Mac)	2.70	2.60	2.45	CALL	CALL
3.5" 2D (For HP)	3.50	3.25	3.15	CALL	CALL

BLACKSHIP COMPUTER SUPPLY
P.O. Box 883362 • San Francisco, CA 94188
Inquiries + No. Cal orders 415-550-0512
So. Cal orders 800-235-6647 x31
USA orders 800-431-6249

Add $3.00 shipping and handling
(CA residents add 6.5% sales tax) VISA/MC/COD

COMPUCAT

Catalog of Computers and Supplies
Our Prices are WHOLESALE +10%

Call, Write or send an E-Letter for our Free Catalog!

We support the complete ATARI and COMMODORE product lines. Ask for our free price list

(408) 353-1836

(Ask about our Free Software.)

Instant shipping (or as fast as we can). *Mastercard & Visa accepted* (no extra charge). Shipping & handling add 6%. California customers add 6.5% sales tax. Order by *phone* (Mon - Fri 10 am - 5 pm PST). Order by *modem* (daily 6 pm-9am) from our online TeleCatalog. Prices subject to change without notice.

COMPUCAT
24500 Glenwood Hwy., Los Gatos, CA 95030

J&R MUSIC WORLD

23 PARK ROW, NEW YORK, N.Y. 10038

ORDER TOLL-FREE 800-221-8180 IN NEW YORK, ALASKA & CANADA CALL (212) 683 0396

Shop at home and save 10% to 50% and more on quality merchandise—fully guaranteed

maxell GOLD STANDARD FLOPPY DISKS

MAXELL MD1D Single sided double density 10-pack $19.95
MAXELL MD2D Double sided double density 10-pack $24.95

NEW commodore 128 128K PERSONAL COMPUTER
Call For Price

'TOP 10' COMPUTER HARDWARE

ATARI 520ST COLOR COMPUTER SYSTEM
- 512K RAM
- MAC-LIKE Graphics Menus Windows

$849.95

COMMODORE 64 COMPUTER	Call
COMMODORE 1541 DISK DRIVE	$179.95
ATARI 800XL COMPUTER	$89.95
ATARI 1050 DISK DRIVE	$169.95
IBM PC PERSONAL COMPUTER	$1799.95
APPLE IIe PRO SYSTEM	$1369.95
COMMODORE 1702 COLOR MONITOR	$189.95
ZENITH ZVM-123 AMBER MONITOR	$99.95
XTRON IA 12K RGB MONITOR	$109.95

'TOP 10' COMPUTER PRINTERS

STAR MICRONICS SG-10 DOT MATRIX COMPUTER PRINTER

$249.95

CANON PW1080A IMPACT MATRIX	$279.95
AXIOM ELITE-5 LETTER QUALITY	$189.95
JUKI 6100 LETTER QUALITY	$369.95
Commodore MPS801 DOT MATRIX	$139.95
ATARI 1027 LETTER QUALITY	$239.95
TOSHIBA P1340 DOT MATRIX	$529.95
OKIDATA 192P DOT MATRIX	$349.95
EPSON SPECTRUM LX-80 MATRIX	$219.95
OKIDATA OKIMATE-10 4-COLOR	$129.95
EPSON FX 185 DOT MATRIX	$499.95

'TOP 10' APPLE SOFTWARE

GATO Spectrum Holobyte	$24.95
FLIGHT SIMULATOR II Sub Logic	$29.95
PFS: FILE Software Pub	$74.95
BANK STREET WRITER Broderbund	$39.95
COMPUTER BASEBALL Micro League	$24.95
PRINTSHOP Broderbund	$34.95
MUSIC CONSTRUCTION SET Elect Arts	$24.95
TYPING TUTOR III Simon & Schuster	$29.95
F-15 STRIKE EAGLE Microprose	$24.95
KARATEKA Broderbund	$24.95

'TOP 10' COMMODORE 64 SOFTWARE

FASTLOAD Epyx	$24.95
SUPER HUEY Cosmi	$14.95
BASEBALL Micro League	$24.95
PRINTSHOP Broderbund	$29.95
GRAPHICS LIBRARY Broderbund	$14.95
SUMMER GAMES II Epyx	$24.95
INDIANA JONES Mindscape	$19.95
PAPERCLIP Batteries Included	$79.95
MACH 5 Access	$24.95
FLIGHT SIMULATOR II Sub Logic	$34.95

'TOP 10' ATARI SOFTWARE

ATARIWRITER Atari	$34.95
SYNFILE PLUS Synapse	$29.95
F-15 STRIKE EAGLE Microprose	$24.95
RESCUE ON FRACTELUS Epyx	$24.95
SARGON II CHESS Hayden	$15.95
BALL BLAZER Epyx	$24.95
PAPERCLIP Batteries Included	$44.95
GHOSTBUSTERS Activision	$19.95
HOMEPAK Batteries Included	$39.95
COMPUTER BASEBALL Micro League	$24.95

'TOP 10' IBM COMPATIBLE SOFTWARE

LOTUS 1-2-3 Lotus Development	$279.95
FLIGHT SIMULATOR Microsoft	$29.95
TURBO PASCAL Borland	$39.95
TYPING TUTOR III Simon & Schuster	$29.95
HITCHHIKERS GUIDE Infocom	$29.95
KINGS QUEST Sierra	$24.95
dBASE III Ashton Tate	$379.95
SYMPHONY Lotus Development	$399.95
FRAMEWORK Ashton Tate	$349.95
MATH BLASTER Davidson Assoc	$29.95

'TOP 10' MACINTOSH SOFTWARE

JAZZ from Lotus Development

$379.95

FACTFINDER Forethought	$119.95
ART GRABBER BODYSHOP Hayden	$34.95
ENSEMBLE Hayden	$169.95
FILEVISION Telos	$129.95
HITCHHIKER'S GUIDE Infocom	$29.95
THINK TANK Living Videotext	$84.95
MUSIC WORKS Hayden	$49.95
TYPING TUTOR II Simon & Schuster	$34.95
MAC VISION Koala Technologies	$199.95

'TOP 10' COMPUTER ACCESSORIES

discwasher SPIKEMASTER SURGE SUPRESSOR

$49.95

SURGE SUPRESSOR Midland/Ross	$24.95
DISK DRIVE CLEANING KIT Head	$14.95
COMPUTER PAPER	$14.95
PARALLEL PRINTER INTERFACE Cardco G-WIZ	$59.95
DISKETTE STORAGE Ring King	$14.95
80-COLUMN PRINTER STAND Ring King	$29.95
SM2400 SMART MODEM Hayes	$629.95
V1660 MODEM Commodore	$59.95
SMALL COMPUTER TABLE Bush	$79.95

All SOFTWARE DISK-BASED UNLESS OTHERWISE NOTED

NOTE: WE CARRY A FULL LINE OF AUDIO, VIDEO AND COMPUTERS IN EVERY MAJOR BRAND. THIS LISTING IS JUST A SMALL SAMPLE IN OUR $10,000,000 INVENTORY. PLEASE CALL US TOLL FREE OR WRITE TO US FOR PRICES ON ITEMS NOT LISTED IN THIS AD

ALL PRODUCTS CARRY U.S. WARRANTIES
ALL DEFECTIVE PRODUCTS EXCHANGED WITHIN 30 DAYS

CUT OUT AND MAIL

ALL NEW FOR 1985

- AUDIO/VIDEO/ELECTRONICS CATALOG
- COMPUTER CATALOG
- RECORD, CASSETTE AND CD CATALOG
- VIDEO MOVIE CATALOG

J&R MUSIC WORLD
4 FREE CATALOGS

PLEASE CHECK WHICH CATALOGS YOU WANT

☐ AUDIO VIDEO ☐ COMPUTER ☐ RECORD CASSETTE ☐ VIDEO MOVIE

23 PARK ROW, DEPT. C11, NYC, NY 10038

NAME ____________
ADDRESS ____________
CITY ________ STATE ____ ZIP ____

CUT OUT AND MAIL

Advertisers Index

Technical

Breakthrough

Number 27

Price and performance no other printer interface can touch!

Only CARDCO would dare improve on its own best seller (the +G has far out-sold any other printer interface, and has set the industry standard by which competitors are judged). The G-Wiz is even better — and costs 30% less.* Our 27th major innovation in Commodore accessories has all the +G's features, and more...

Built-In Buffer for More Speed
The G-Wiz buffer dumps high-resolution screens up to 18 times faster than competitive interfaces without buffers. No more waiting! A complex 50-minute printout now takes just 2.5 minutes with the G-Wiz.

Exclusive! Aspect Ratio Selection
Only the G-Wiz matches graphics characters to standard characters on Okidata, Epson, Star Micronics, Prowriter, Smith Corona, C-ITOH, Gorilla Banana, and many other dot matrix printers. Now you can perfectly align high resolution graphics characters within text blocks, or in columns.

CARDCO excellence triumphs again! The G-Wiz is the "best bang for the buck" on the printer interface market today — and it's backed by CARDCO'S exclusive lifetime warranty! G-Wiz: another distant target for the competition to shoot at.

CARDCO, Inc. 300 S. Topeka/Wichita, KS 67202

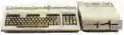
* Actual price may vary according to dealer and region

The Wizards from the Land of Oz Have Done It Again!

HOW TO EVOLVE TO A HIGHER INTELLIGENCE.

THE COMMODORE 128.

The first step is buying the Commodore 128™ Personal Computer. The smartest computer available for the price. It's like getting three computers for less than one usually costs. You can run CP/M® business software, the new programs written for the 128, and over 3,000 Commodore 64® programs. You start out with more software than most machines give you after years on the market.



THE COMMODORE 128 WORKS FASTER.

To run all that software and run it faster, you'll want the 1571 Disk Drive. You can't find a faster drive at the price. It transfers nearly 1,000 words a second (5200 cps), so you can load most programs instantly.

THE COMMODORE 128 GETS SMARTER.

Now try improving your memory. Plug in our 1750 RAM Expansion Module and your 128 moves up to a powerful 512K. That's enough to handle just about anything you can dish out, from complicated business forecasting to giant data bases.

THE COMMODORE 128 LEARNS TO COMMUNICATE.

There's no real intelligence without the ability to communicate. So you'll want our 1670 Modem/1200. It puts you in touch with a new world of shopping, banking, communications and information over your telephone line. And it operates at a lightning-fast 1200 baud to save on your phone bill.

THE COMMODORE 128 LEARNS TO WRITE.

Looking good in print could be your next move with the MPS 1000 Printer. It's a new dot matrix printer designed to make the most of the 128's speed and high-resolution graphics. The MPS turns out about 1200 words a minute (100 cps) of draft-quality printing, or gives you near-letter-quality at about 240 words a minute (20 cps).

THE COMMODORE 128 IMPROVES YOUR VISION.

Brains aren't enough without good looks, so improve your vision with Commodore's new 1902 RGB Color Monitor. The high-resolution screen gives you a sharper image and better color than your standard TV, so you can really appreciate the 128's great graphics.

All these evolutionary steps ahead won't set you back when it comes to paying for them. Additions to your Commodore 128 are available at a store near you and are as affordable as the 128 itself. We think that's a smart way to help you build a computer system.

COMMODORE 128 PERSONAL COMPUTER
A Higher Intelligence